QuickStart Instructions

Linux-Kit

phyCORE-MPC5200B IO

Using Eclipse and the GNU Cross Development Toolchain

Note: The PHYTEC MPC5200IO-Disc includes the electronic version of the phyCORE-MPC5200B IO English Hardware Manual

Edition: August 2009

A product of a PHYTEC Technology Holding company

In this manual copyrighted products are not explicitly indicated. The absence of the trademark (™) and copyright (©) symbols does not imply that a product is not protected. Additionally, registered patents and trademarks are similarly not expressly indicated in this manual.

The information in this document has been carefully checked and is believed to be entirely reliable. However, PHYTEC Messtechnik GmbH assumes no responsibility for any inaccuracies. PHYTEC Messtechnik GmbH neither gives any guarantee nor accepts any liability whatsoever for consequential damages resulting from the use of this manual or its associated product. PHYTEC Messtechnik GmbH reserves the right to alter the information contained herein without prior notification and accepts no responsibility for any damages which might result.

Additionally, PHYTEC Messtechnik GmbH offers no guarantee nor accepts any liability for damages arising from the improper usage or improper installation of the hardware or software. PHYTEC Messtechnik GmbH further reserves the right to alter the layout and/or design of the hardware without prior notification and accepts no liability for doing so.

	EUROPE	NORTH AMERICA
Address:	PHYTEC Technologie Holding AG Robert-Koch-Str. 39 55129 Mainz GERMANY	PHYTEC America LLC 203 Parfitt Way SW, Suite G100 Bainbridge Island, WA 98110 USA
Ordering Information:	+49 (800) 0749832 order@phytec.de	1 (800) 278-9913 sales@phytec.com
Technical Support:	+49 (6131) 9221-31 support@phytec.de	1 (800) 278-9913 support@phytec.com
Fax:	+49 (6131) 9221-33	1 (206) 780-9135
Web Site:	http://www.phytec.de	http://www.phytec.com

6th Edition August 2009

5 min

35 min

70 min

20 min

1 Introduction

5 min

In this QuickStart you can find general information information on the PHYTEC phyCORE®-MPC5200B IO and an overview of the Eclipse software development tool and GNU GCC C/C++ Cross-Development Toolchain. You can also find instructions on how to run example programs on the phyCORE®-MPC5200B IO, mounted on the PHYTEC phyCORE® Development Board MPC5200B IO, in conjunction with the Eclipse development tool.

Please refer to the phyCORE-MPC5200B IO Hardware Manual for specific information on such board-level features as jumper configuration, memory mapping and pin layout.

1.1 Rapid Development Kit Documentation

This "Rapid Development Kit" includes the following electronic documentation on the enclosed "PHYTEC Linux-PowerPC-Disc":

- PHYTEC phyCORE-MPC5200B IO Hardware Manual and Development Board Hardware Manual
- MPC5200B IO controller User's Manuals and Data Sheets
- this QuickStart Instruction with general "Rapid Development Kit" description, software installation advice and an example program enabling quick out-of-the box start-up of the phyCORE®- MPC5200B IO in conjunction with the Eclipse / GNU GCC C/C++ software development toolchain

1.2 Professional Support Packages available

This Kit comes with free installation support. If you do have any questions concerning installation and setup, you are welcome to contact our support department.

For more in-depth questions, we offer a variety of custom tailored packages with different support options (e-mail, phone, direct contact to the developer) and different reaction times.

Please contact our sales team to discuss the appropriate support option if professional support beyond installation and setup is important to you.

1.3 Overview of this QuickStart Instruction

This QuickStart Instruction gives a general "Rapid Development Kit" description, as well as software installation advice and an example program enabling quick out- of-the box start-up of the phyCORE®-MPC5200B in conjunction with the Eclipse IDE and GNU GCC/C++ software tools. It is structured as follows:

1) The *"Getting Started"* section describes the configuration of the host platform and the setup to install the tools used in this QuickStart.
2) The *"Getting More Involved"* section provides step-by-step instructions on how to configure and build a new kernel, modify the example, create and build new projects and copy output files to the phyCORE®-MPC5200 using Eclipse.
3) The *"Debugging"* section provides information on how to debug an application with the Eclipse debugging interface.

In addition to the dedicated data for this Rapid Development Kit, the PHYTEC MPC5200IO-Disc contains supplemental information on embedded microcontroller design and development.

1.4 Conventions used in this QuickStart

The following is a list of the typographical conventions used in this book:

Italic Used for file and directory names, program and command names, command-line options, menu items, URLs, and other terms that corresponds the terms on your desktop.

Bold Used in examples to show commands or other text that should be typed literally by the user.

Pay special attention to notes set apart from the text with the following icons:

At this part you might leave be path of this QuickStart.

This is a warning. It helps you to avoid annoying problems.

You can find useful supplementary information about the topic.

At the beginning of each chapter you can find information of the time to pass the following chapter.

You have successfully passed an important part of this QuickStart.

You can find information to solve problems.

1.5 System Requirements

Use of this "Rapid Development Kit" requires:

- the PHYTEC phyCORE-MPC5200B IO,
- the PHYTEC Development Board with the included DB-9 serial cable, AC-to-DC adapter supplying 12 V DC/min. 1.5 A,
- PHYTEC Distribution based on OSELAS from Pengutronix
- an IBM-compatible host-PC (586 or higher)
- openSUSE Linux 11.0 OSS (x86**)** and KDE desktop
- recommended free disk space: 2 GB

For more information and example updates, please refer to the following sources:

http://www.phytec.de
support@phytec.de

http://www.freescale.com/

1.6 The PHYTEC phyCORE-MPC5200B IO

The phyCORE-MPC5200B IO represents an affordable yet highly functional Single Board Computer (SBC) solution in the size 53 x 57 mm^2 for the tiny-Board and 57 x 84 mm^2 for the IO-Board . The standard board is populated with a Freescale PowerPC Microcontroller MPC5200B featuring a 32-bit processor architecture with Double precision FPU, 396 MHz processor speed, Peripheral component interconnect (PCI) controller, ATA controller, BestComm DMA subsystem, 6 programmable serial controllers (PSC) configurable for the following functions:
Fast Ethernet controller (FEC), Universal serial bus controller (USB revision 1.1 host), Two inter-integrated circuit interfaces (I2C), Serial peripheral interface (SPI), Dual CAN 2.0 A/B controller (MSCAN), J1850 byte data link controller (BDLC)

All applicable LocalPlus data/address lines and applicable signals extend to two high-density 100-pin Molex SMT pin header connectors (pin width is 0.635 mm./25mil) lining the circuit board edges. This enables the phyCORE-MPC5200B tiny/IO to be plugged like a “big chip” into target hardware.

The standard memory configurations of the phyCORE-MPC5200B tiny features 64MByte DDR-SDRAM and 16MByte external Flash. The standard memory configurations of the phyCORE-MPC5200B IO features 128MByte DDR-SDRAM and 32MByte external Flash. The external Flash supports direct on-board programming without additional programming voltages.

phyCORE-MPC5200B tiny Technical Highlights

- phyCORE dimensions 53 x 57 mm with two high-density 100-pin Molex SMT pin header connectors
- Processor: Freescale Embedded PowerPC MPC5200B, 396 MHz clock

phyCORE-MPC5200B IO Technical Highlights

- phyCORE dimensions 57 x 84 mm^2 with two high-density 200-pin Molex SMT pin header connectors
- Processor: Freescale Embedded PowerPC MPC5200B, 396 MHz clock
- Altera FPGA with up to 8000 Logic Cells
- Battery buffer able SRAM

Internal Features of the MPC5200B:

- e300 core
  - 760 MIPS at 400 MHz (-40 to +85 °C)
  - 32 k instruction cache, 32 k data cache
  - Double precision FPU
  - Instruction and data MMU
- SDRAM / DDR SDRAM memory Interface
  - up to 132 MHz operation
  - SDRAM and DDR SDRAM support
  - 256 MByte addressing range per CS, two CS available
- Flexible multi-function external bus interface
- Peripheral component interconnect (PCI) controller
- ATA controller
- BestComm DMA subsystem
- 6 programmable serial controllers (PSC), configurable for the following functions:
- Fast Ethernet controller (FEC)
  - Supports 100Mbps IEEE 802.3 MII, 10 Mbps IEEE 802.3 MII
- Universal serial bus controller (USB)
  - USB revision 1.1 host
- Two inter-integrated circuit interfaces (I2C)
- Serial peripheral interface (SPI)

- Dual CAN 2.0 A/B controller (MSCAN)
- J1850 byte data link controller (BDLC)
- Test/debug features
- JTAG (IEEE 1149.1 test access port)
- Common on-chip processor (COP) debug port

The PHYTEC Development Board, in card dimensions (195 x 170 mm^2/7.68 x 6.7 in^2.), is fully equipped with all mechanical and electrical components necessary for the speedy and secure insertion of PHYTEC phyCORE-MPC5200B IO Single Board Computer.

Development Board Technical Highlights

- reset push button
- one software programmable LED for processor
- one software programmable LED for fpga
- LEDs for supply voltage monitoring
- power supply for regulated input voltage of +12V. It supplies regulated +3.3 V for the phyCORE-MPC5200B IO and futher supply voltages.
- two DB-9 sockets for the RS-232 interface
- two DB-9 plugs for two separate CAN interfaces
- Ethernet socket with integrated USB Host Interface
- Compact Flash Card Socket
- PCI Socket
- ATA-Interface

1.7 Software Development Toolchains

1.7.1 Eclipse

The Eclipse Platform provides support for C/C++ development. Because the Eclipse Platform is only a framework for developer tools, it doesn't support C/C++ directly; it uses external plug-ins for support. This QuickStart shows you how to make use of the CDT -- a set of plug-ins for C/C++ development in conjunction with the GNU GCC C/C++ Toolchain.

The CDT is an open source project (licensed under the Common Public License) implemented purely in Java as a set of plug-ins for the Eclipse SDK Platform. Theese plug-ins add a C/C++ Perspective to the Eclipse Workbench that can now support C/C++ development with a number of views and wizards, along with advanced editing and debugging support.

Due to its complexity, the CDT is broken down into several components, which take the form of separate plug-ins. Each component operates as an autonomous project, with its own set of committers, bug categories, and mailing lists. However, all plug-ins are required for the CDT to work properly. Here is a list of the plug-ins/components:

- **Primary CDT plug-in** is the "framework" CDT plug-in.
- **CDT Feature Eclipse** is the CDT Feature Component.
- **CDT Core** provides Core Model, CDOM, and Core Components.
- **CDT UI** is the Core UI, views, editors, and wizards.
- **CDT Launch** provides the launch mechanism for external tools such as the compiler and debugger.
- **CDT Debug Core** provides debugging functions.
- **CDT Debug UI** provides the user interface for the CDT debugging editors, views, and wizards.
- **CDT Debug MI** is the application connector for MI-compatible debuggers.

1.7.2 The Gnu Cross Development Toolchain

Cross-development in general refers to the overall software development process that produces a single application or a complete system running on a platform that is different from the development platform. This is an important concept when the target system doesn't have a native set of compilation tools, or when the host system is faster and has greater resources.

The platform where the actual development takes place is called the host platform. The platform where the final application is tested and run is called target platform. In this QuickStart we are using a x86-architecture based Linux as host platform. As target platform we are using the arm architecture with a MPC5200B tiny/IO CPU.

Building a program for a CPU architecture different from the one used on the machine where the compilation is done, is accomplished using a cross-compiler toolchain and cross-compiled libraries. In this QuickStart we are using the GNU C/C++ Cross Development Toolchain.

2 Getting Started

35 min

In this chapter you will establish the basis to pass the steps in this QuickStart. First you will learn how to configure the host platform. You will install additional software packages and setup the network configuration for connecting your host to the target. After connecting the host to the target you will copy an application to the target. At the end of this chapter you will be able to start a first demo application on the target.

2.1 Requirements of the Host Platform

To pass the following steps in this Quick Start, you will need a host PC with an installation of openSUSE 11.0 (x86) and the KDE desktop.
When you are installing openSUSE 11.0, you can select *KDE* as *Desktop selection*. The default packages to use openSUSE 11.0 with your host PC will be selected automatically. This default selection will suffice to pass the steps in these Quick Start Instructions. The installation of additional packages and configurations will be described on the following pages.
In the following configuration steps we assume that the host PC is not connected to any other network. The target and host will be connected with a cross-over cable via a peer-to-peer connection. If your host is part of a company's network, we recommend disconnecting your host from such a network.
In these Quick Start Instructions you will have to shutdown the firewall and configure the network card of your host PC. If your host PC is connected to another network, changing the IP address can cause conflicts with existing hosts.

2.2 Configuring the Host Platform

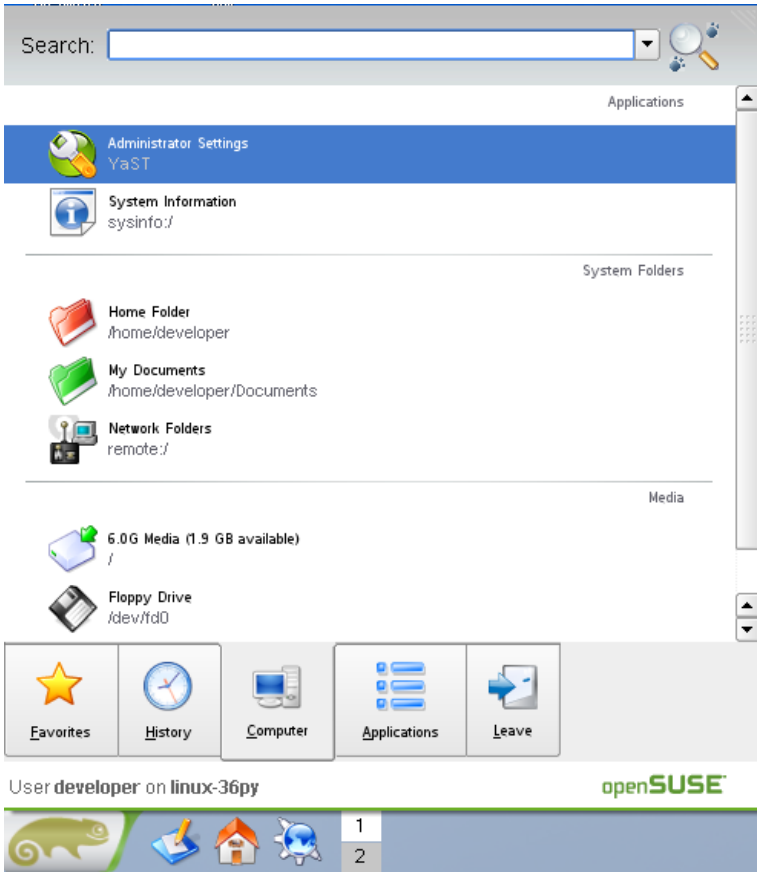
In this passage you will learn how to configure the host platform. You will execute the following steps:

- Install additional software packages. These packages are necessary to accomplish the steps in the Quick Start Instructions.
- Set up the network configuration to use the host PC with your target.
- Disable the firewall. If the firewall is enabled, you will have problems with connecting to the target.
- Set up a TFTP server. You can use a TFTP server to download files (e.g. kernel and root file system images) to the target from within the target's boot loader.

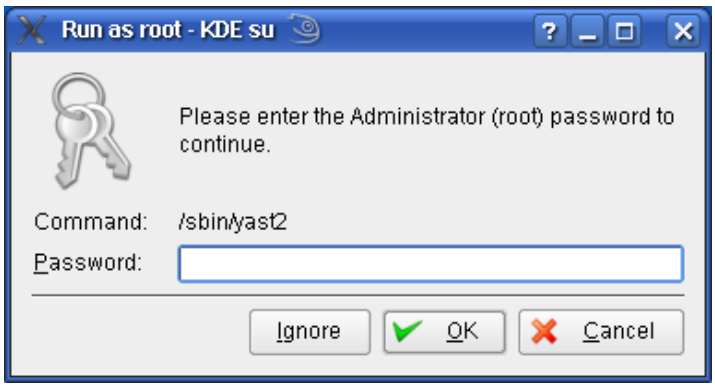
2.2.1 Installing Software Packages

To accomplish the steps in the Quick Start Instructions, you will have to install additional packages.

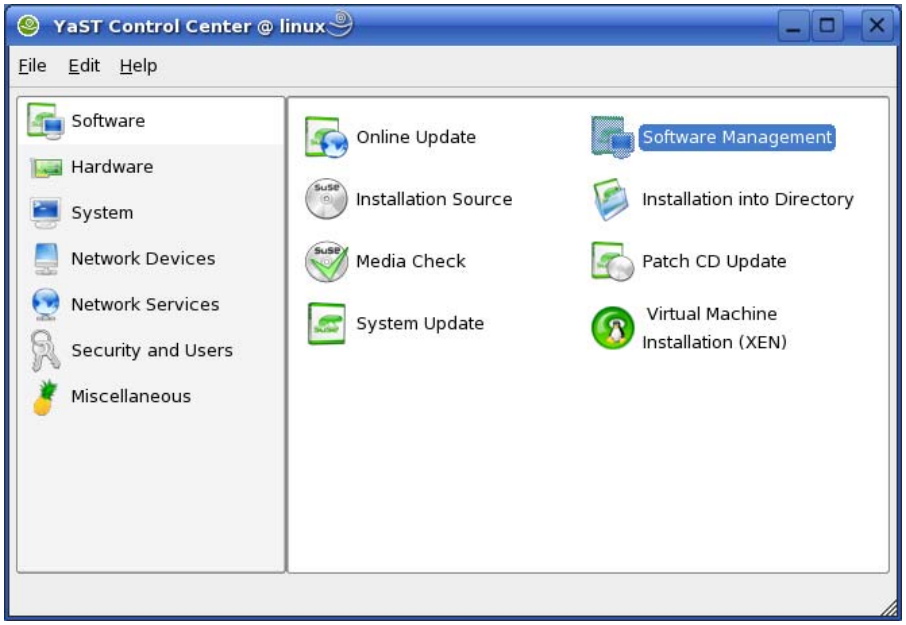
If you don't install all of these packages, the setup may fail or some configuration steps won't work correctly.



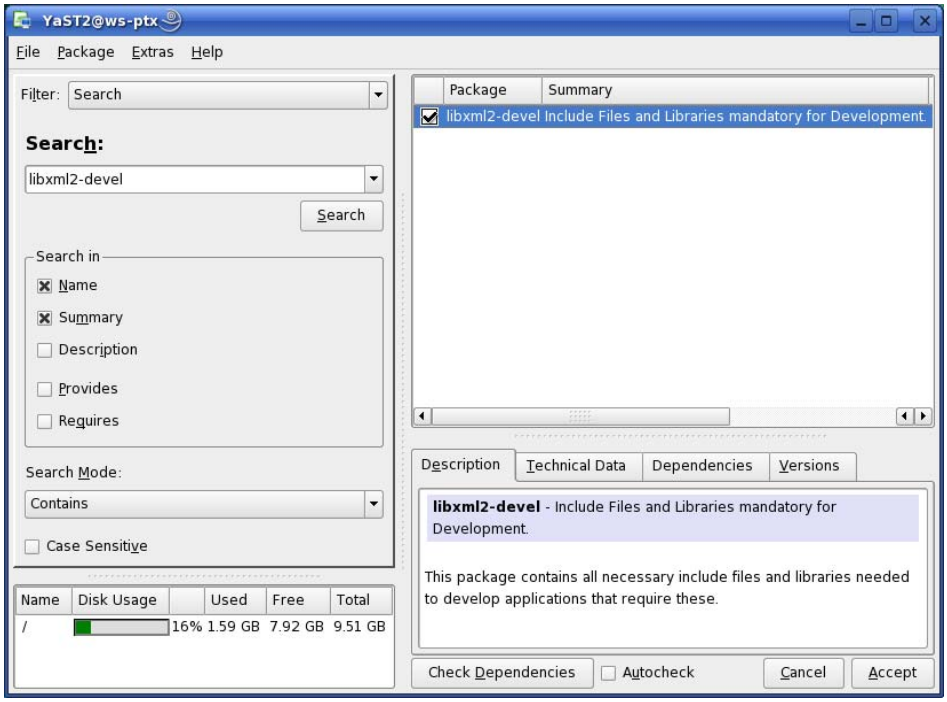
- Open the *K menu* from the lower-left corner of the desktop and click on the tab *Computer*.
- Open the *Administrator Settings / YaST*.



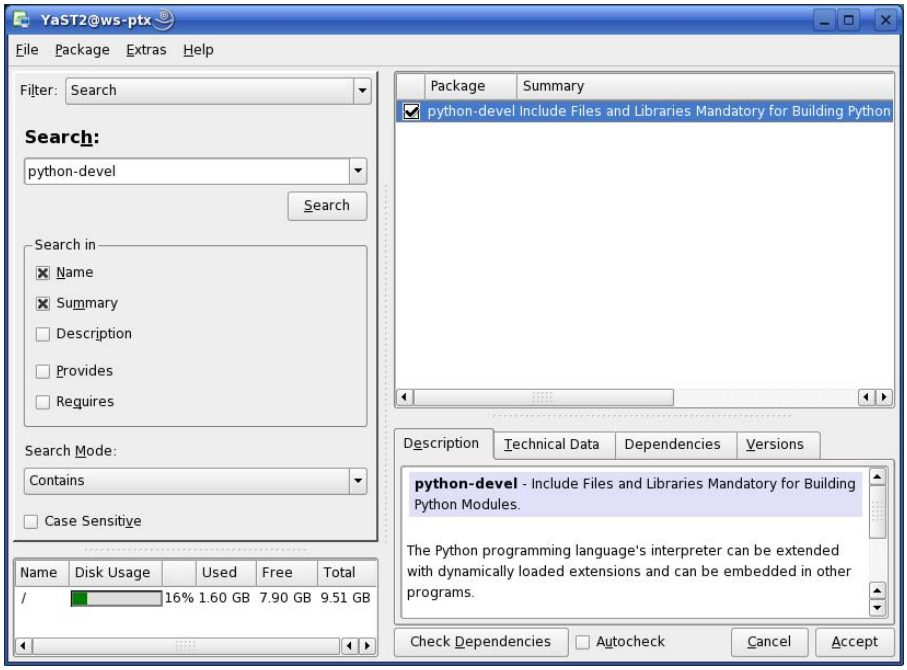
- Enter your root password and click *OK*.



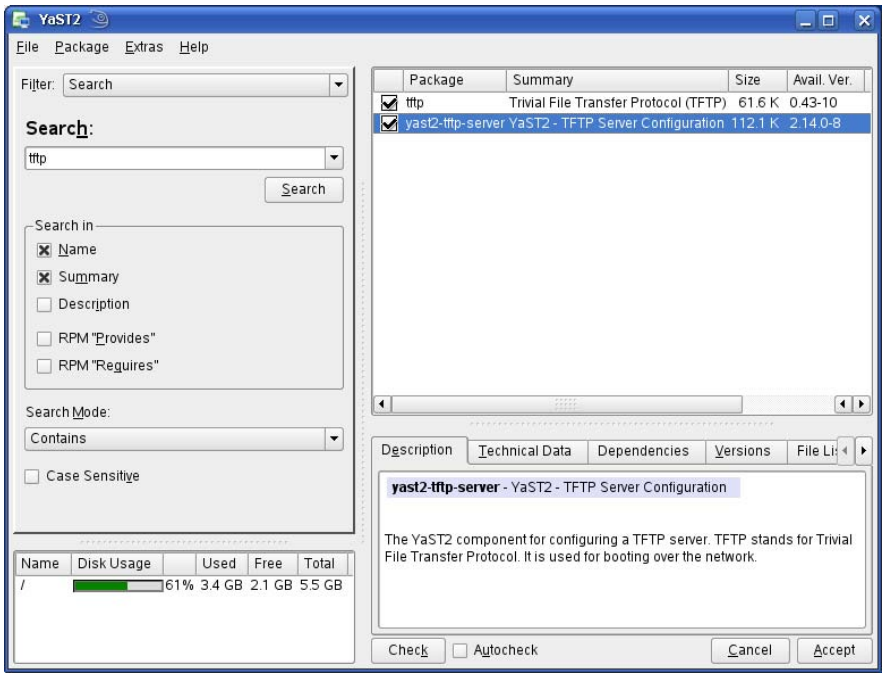
- Open *Software Management* in *Software*.



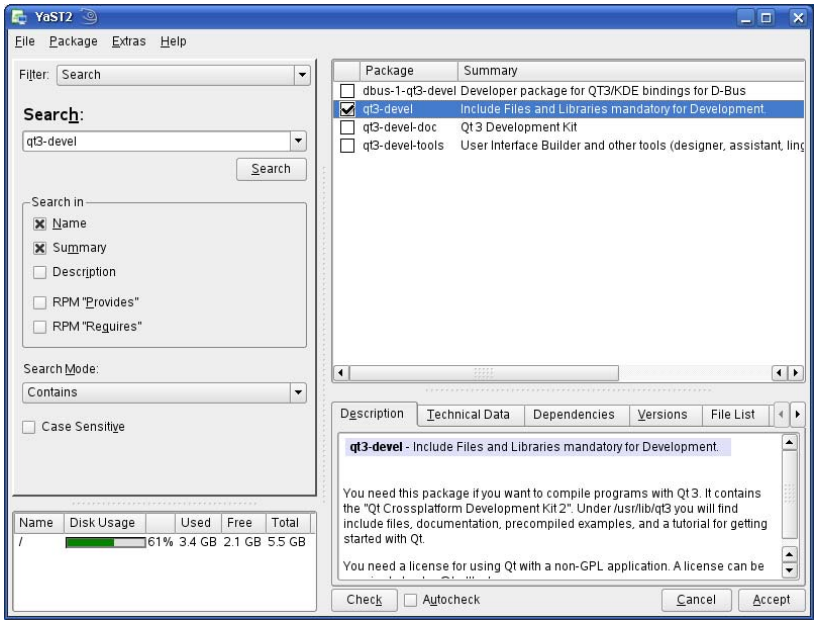
- Select the filter *Search*.
- Type **libxml2-devel** and click the *Search* button.
- Check *libxml2-devel.*



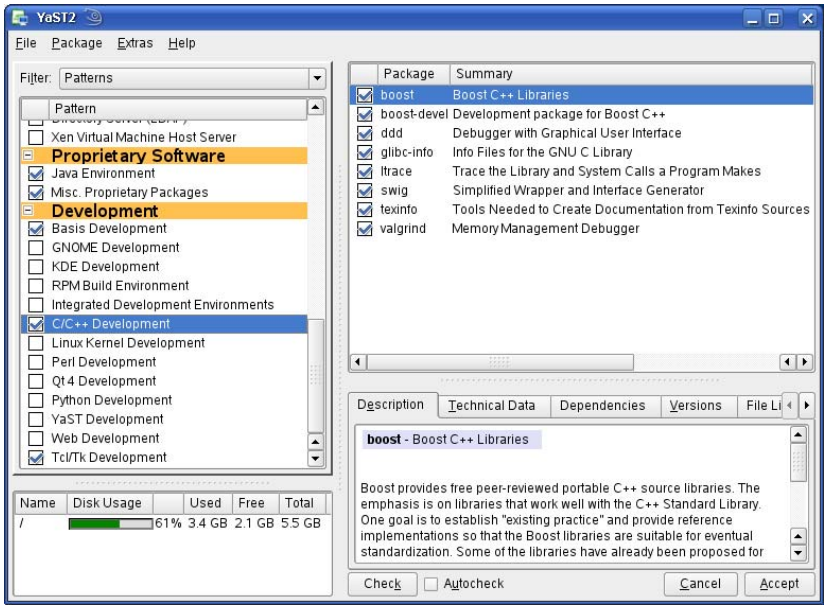
- Type **python-devel** and click the *Search* button.
- Check *python-devel*.



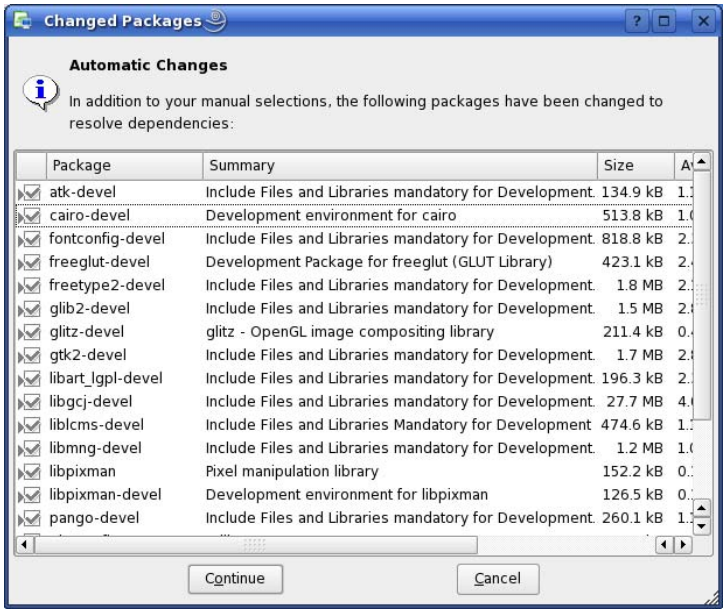
- Type **tftp** and click the *Search* button.
- Check the packages *tftp* and *yast2-tftp-server*.



- Type **qt3-devel** and click the *Search* button.
- Check *qt3-devel.*



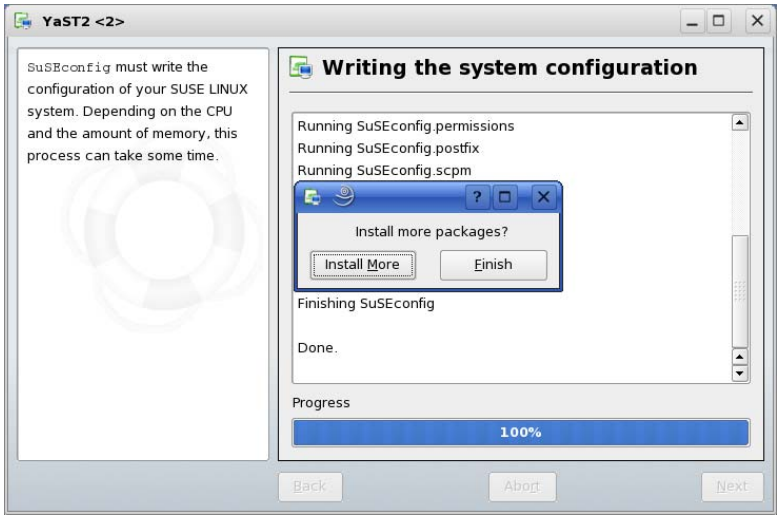
- Select the filter *Patterns*.
- Select *Base Development*, *C/C++ Development*, and *Tcl/Tk Development*.
- Click *Accept*.



Some additional packages will be selected automatically to resolve any dependencies.

If problems occur while resolving dependencies, we recommend going back to a default configuration.

- Click *Continue* to install the packages.

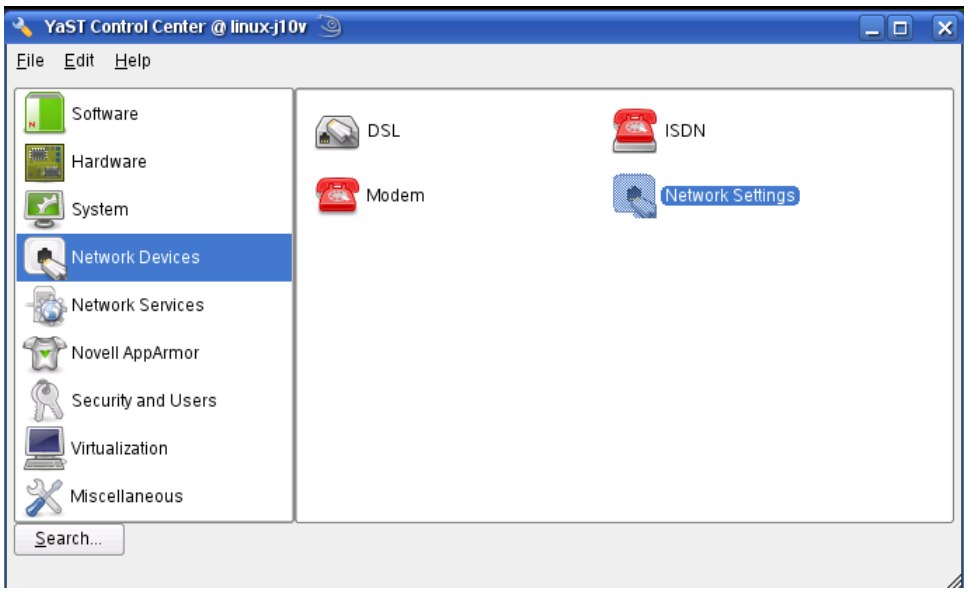


- Click *Finish*.

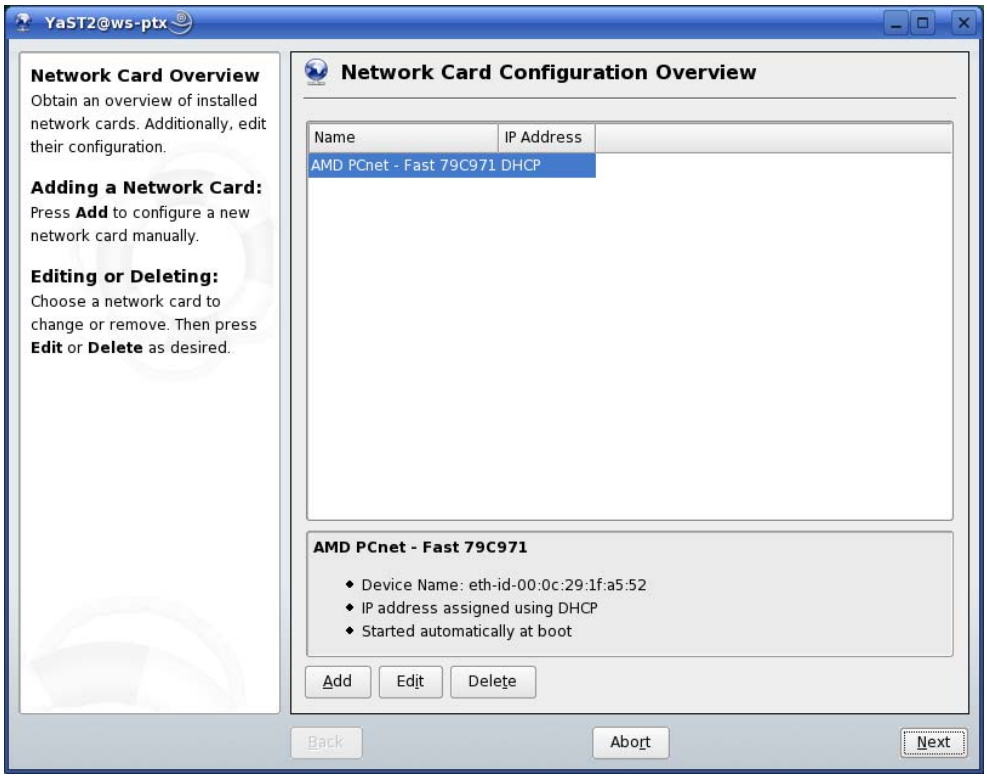
2.2.2 Set Up Network Card Configuration

In the following steps you will have to configure the IP address of your host. We recommend disconnecting your host from any other network. If you change the host's IP, chances are that problems may occur with other hosts in the network.

- Open the *YaST Control Center* if it is not already opened.



- Choose *Network Settings* in *Network Devices*.

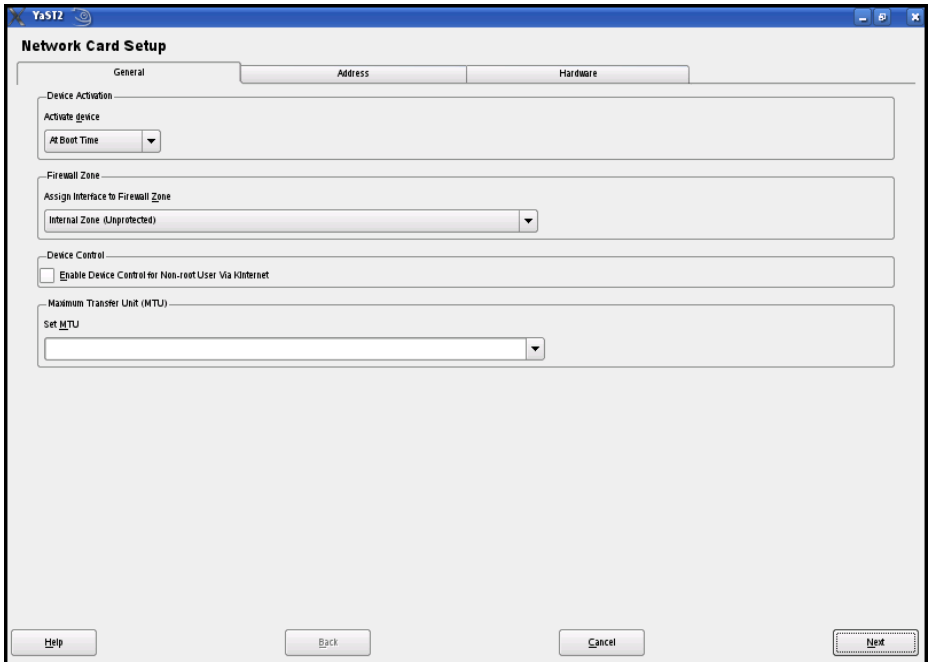


- Select the right network card (if more than one network card is installed on your host).
- Click *Edit* to enter the *Network Card Setup.*
- Choose *Static address setup*.
- Enter IP address **192.168.3.10** and subnet mask **255.255.255.0**

2.2.3 Disabling the Firewall

To ensure that there are no problems with connections to the target, the host's firewall should be disabled.

- Select the *General* tab in the upper-left corner.



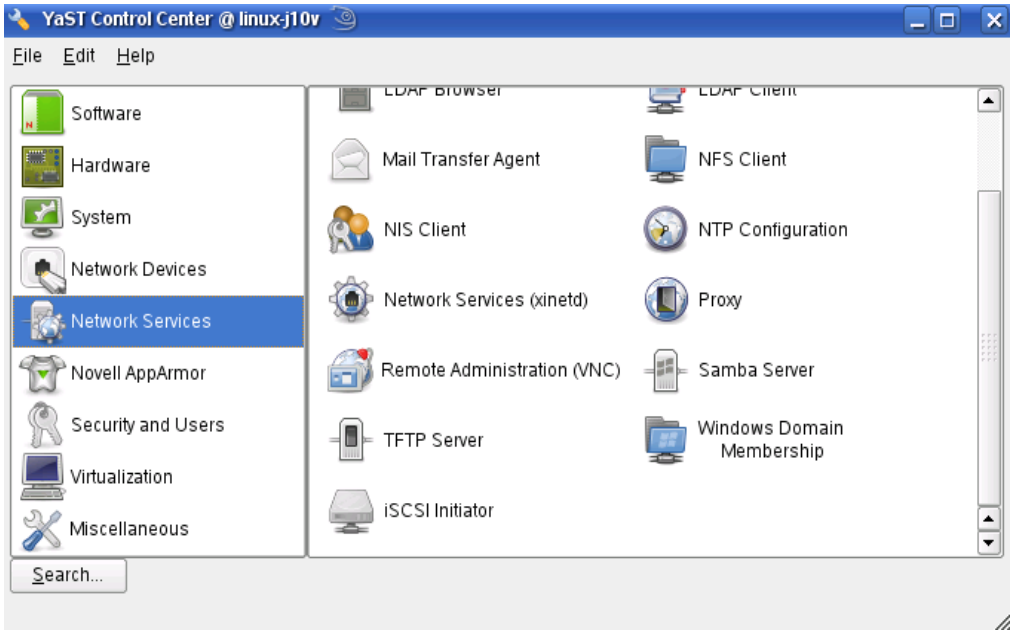
- Use the drop-down box in the *Firewall Zone* settings to set the current interface to *Internal Zone (Unprotected).*
- Then press *Next,* and in the following window click *Finish* to complete the settings.

The firewall is now disabled for this network card.

2.2.4 Set Up TFTP Server

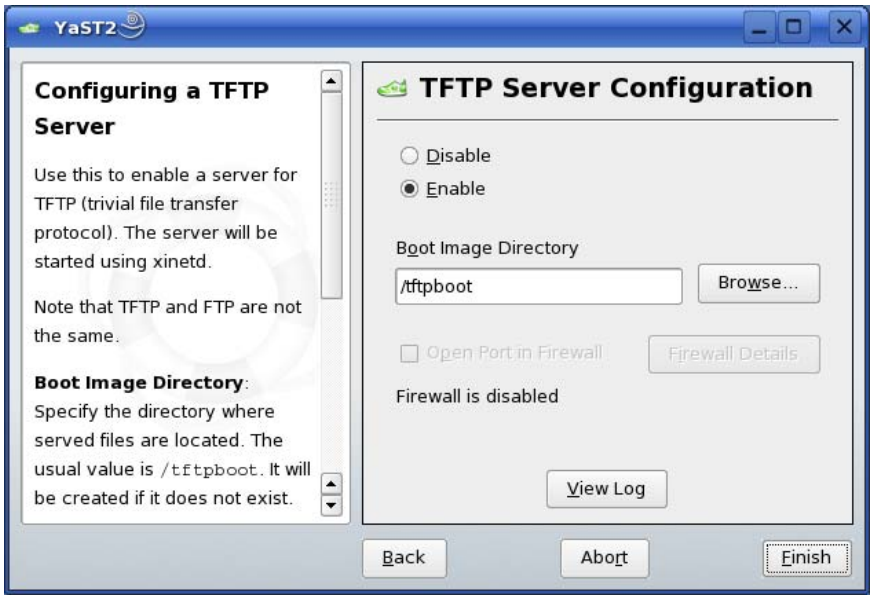
Later in this Quick Start you will learn how to write a new kernel image into the flash memory of the target. To download the kernel image from the target, you need have to have a TFTP server running. In this passage we show you how to configure a TFTP server.

- Open the *YaST Control Center* if it is not already opened.

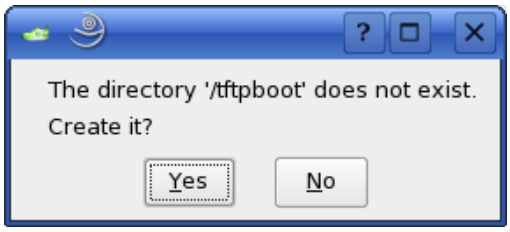


- Choose *TFTP Server* in *Network Services*.

If the *TFTP Server* icon does not exist, restart the YaST Control Center.



- Switch the selection to *Enable*.
- The path of the boot image directory should be */tftpboot*. If there is a different path, change it to */tftpboot.*
- Click *Finish*.



- Click *Yes* to create the */tftpboot* directory.

  The TFTP server will be started.
- Close the YaST Control Center.

You have successfully finished the configuration of the host platform.

2.3 Linux-MPC5200IO-Kit Setup

In this section you will find a description of the Linux-PowerPC-Kit setup. The whole setup will be done by a graphical interface. At the end of the setup, you will find all programs to develop applications for the target on your host PC.

The setup contains the following programs:

- GNU C/C++ Cross Development Toolchain - you can use the toolchain to develop programs for the target on your host PC.
- Eclipse SDK with CDT – the Eclipse SDK is a platform and application frameworks for building software, which can use the GNU C/C++ Cross Development Toolchain.
- Komport - a program for serial communication with the target.
- Linux Kernel archive– this kernel archive contains patches to compile the kernel for the phyCORE®MPC5200.
- HelloWorld program - this program can be used to test downloading and executing a program on the target.
- mkimage – this program will be used to create the kernel image file for the target.

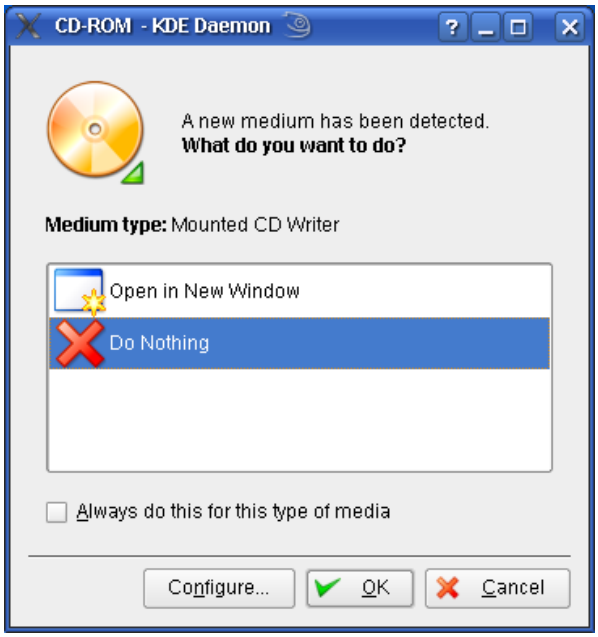
There will be some additional configuration on your host PC:

- The setup program will create desktop links to the installed programs.
- The setup will also create desktop links to access the target with FTP and telnet.
- The path of the cross development toolchain will be added to the environment path.
- Read / Write access to the serial interface will be added to the user rights to use the serial communication program Komport.
- The setup will configure Komport.

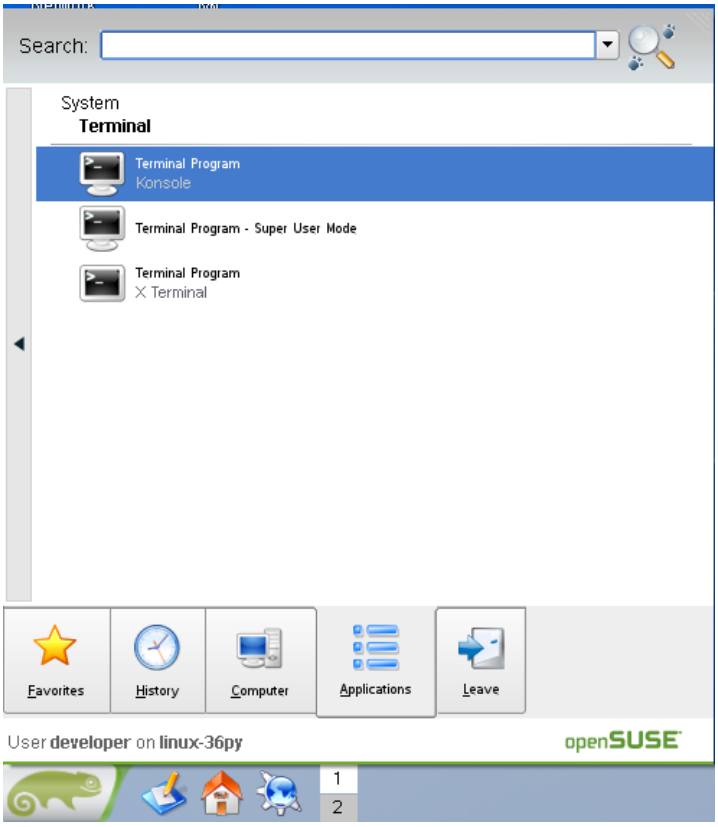
Starting the Setup:

- To start with the Linux-MPC5200IO-Kit Setup enter your PHYTEC Linux-MPC5200IO-Disc into your CD-ROM drive.

The following dialog may appear:

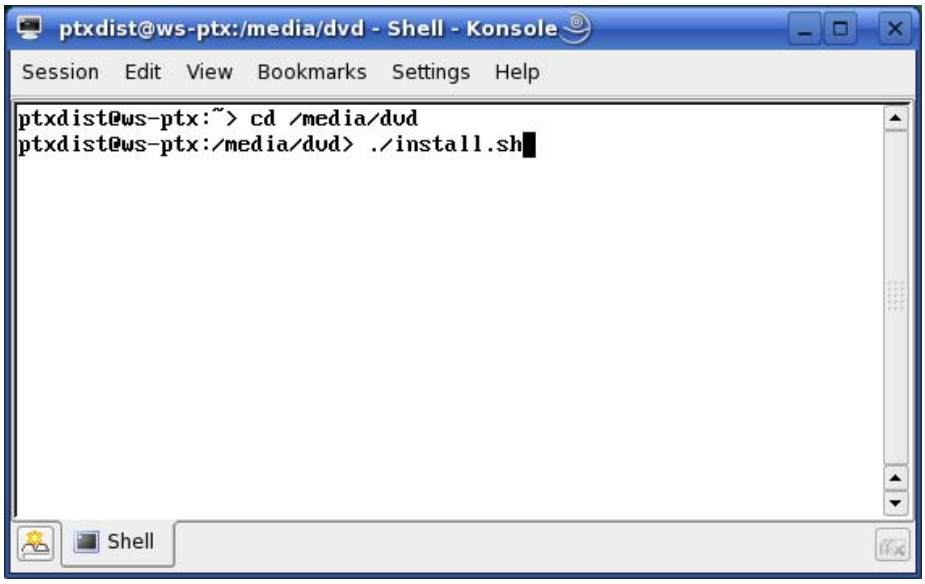


- Click *Cancel*.



- From the *K menu*, select the *Applications* tab.
- Select *System* ► *Terminal* ► *Terminal Program / Konsole*.

In the next step you will have to change to the directory accordant to your drive. If you use a dvd drive change to the directory */media/dvd*. If you have installed a cdrom drive in your host, you will have to change to the directory */media/cdrom*.



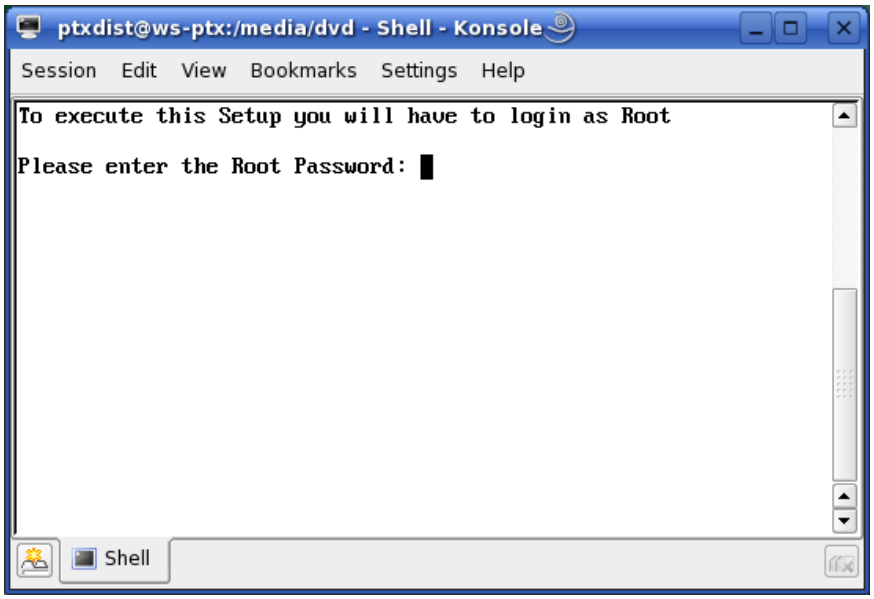
- Type

  cd /media/cdrom

  or

  cd /media/dvd

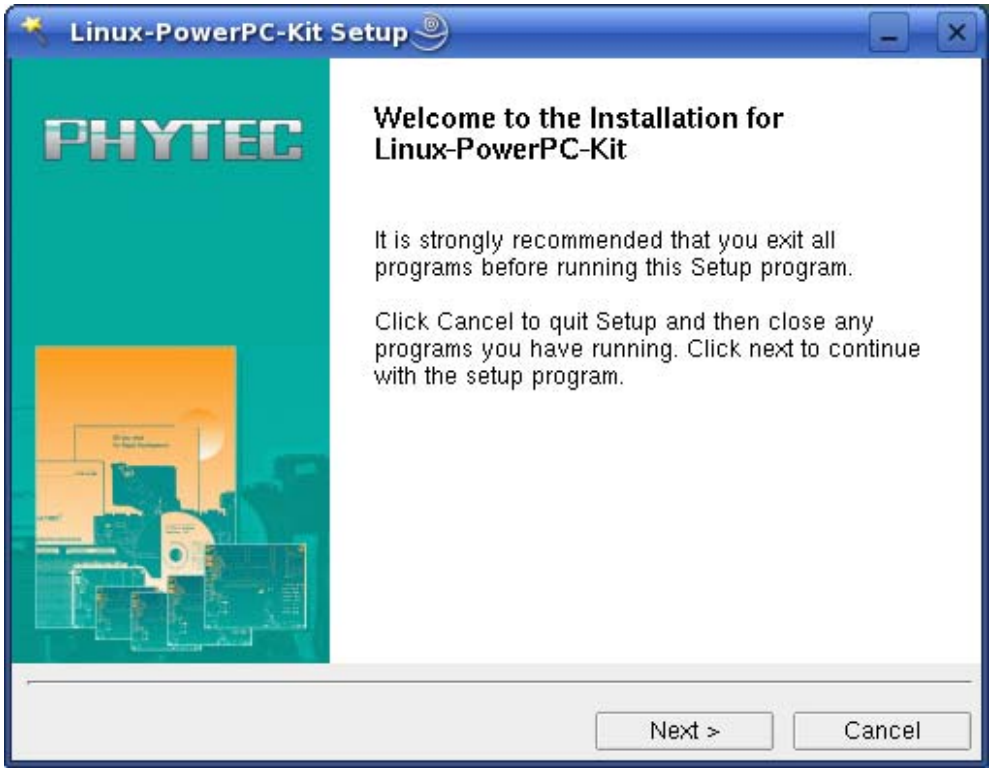
  accordant to your drive.
- Enter **./install.sh** to start the setup program.



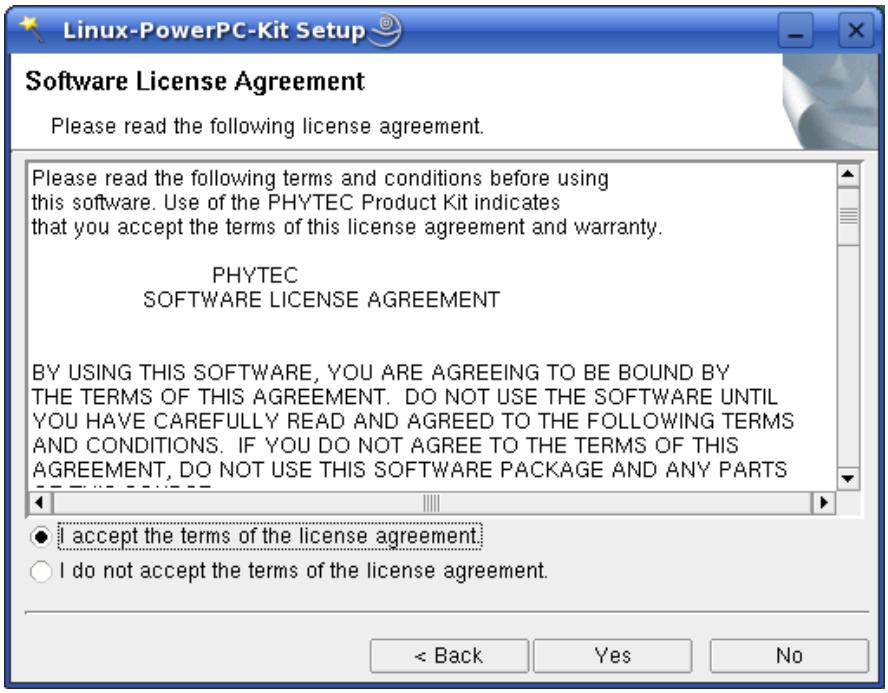
- Enter the root password.

- Click on *Yes* to proceed.

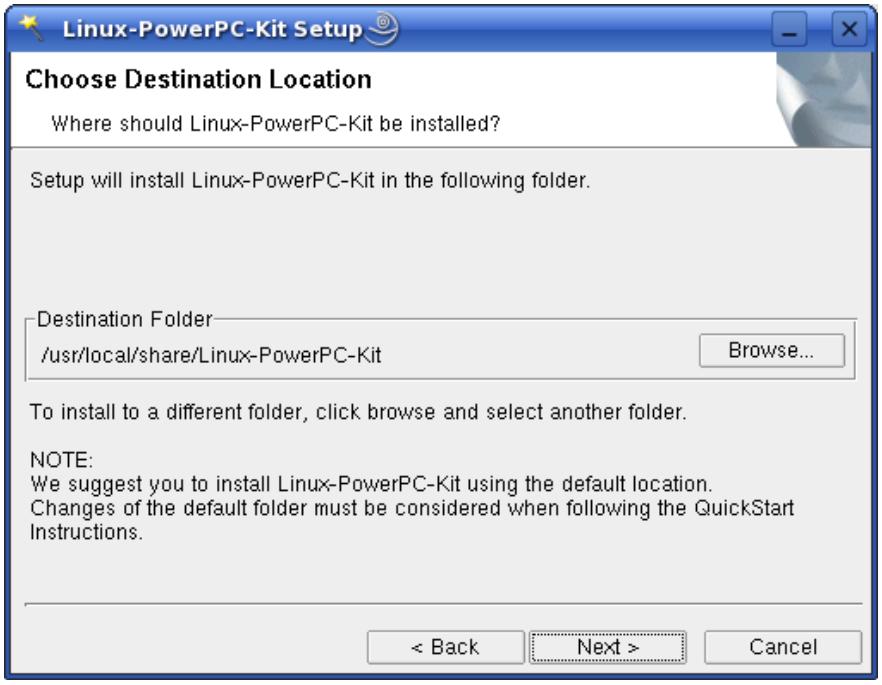
The welcome screen appears.



- Click on *Next* to continue.



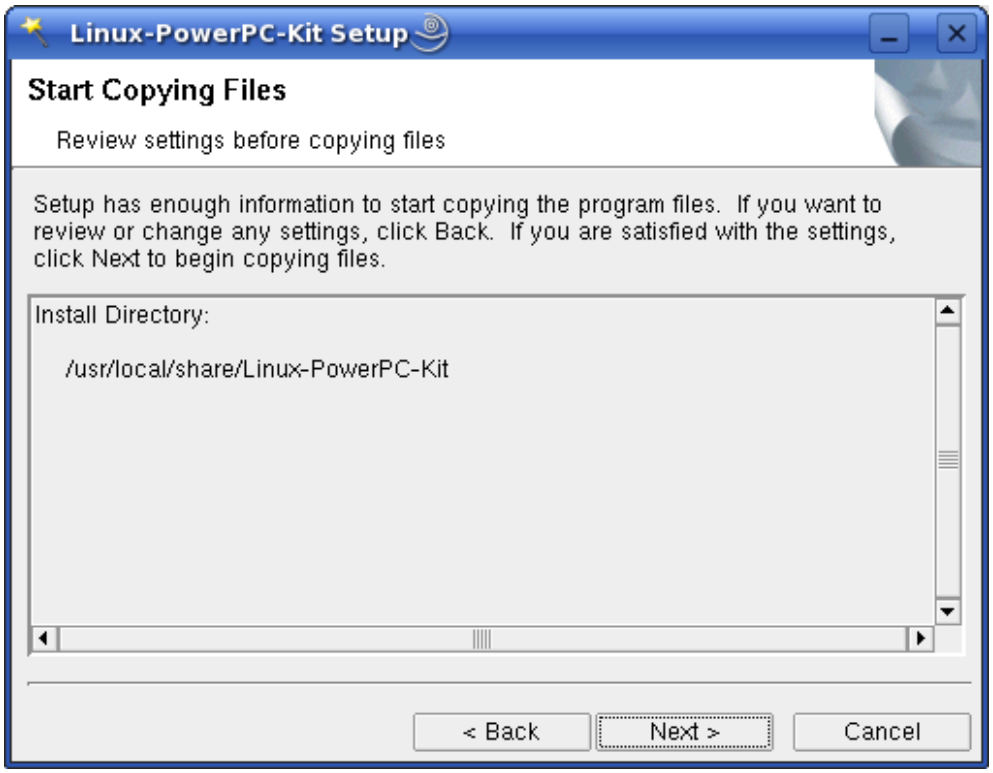
- Choose *I accept the terms of the license agreement.*
- Click on button *Yes.*



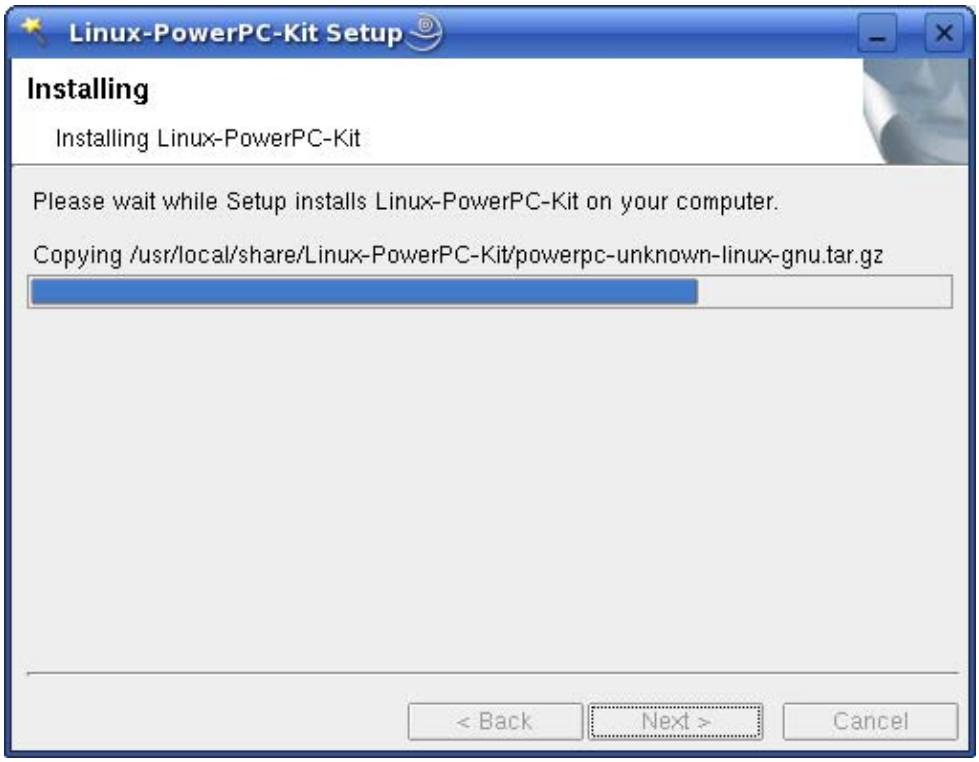
- Click on *Next* to continue.

The default destination location is */usr/local/share/Linux-MP5200IO-Kit* . All path and file statements within this QuickStart Instruction are based on the assumption that you accept the default install paths and drives. If you decide to individually choose different paths you must consider this for all further file and path statements when working with this QuickStart.

We strongly recommend accepting the default destination location.



- Click on *Next* to install the setup files to */usr/local/share/MPC5200IO-Kit.*



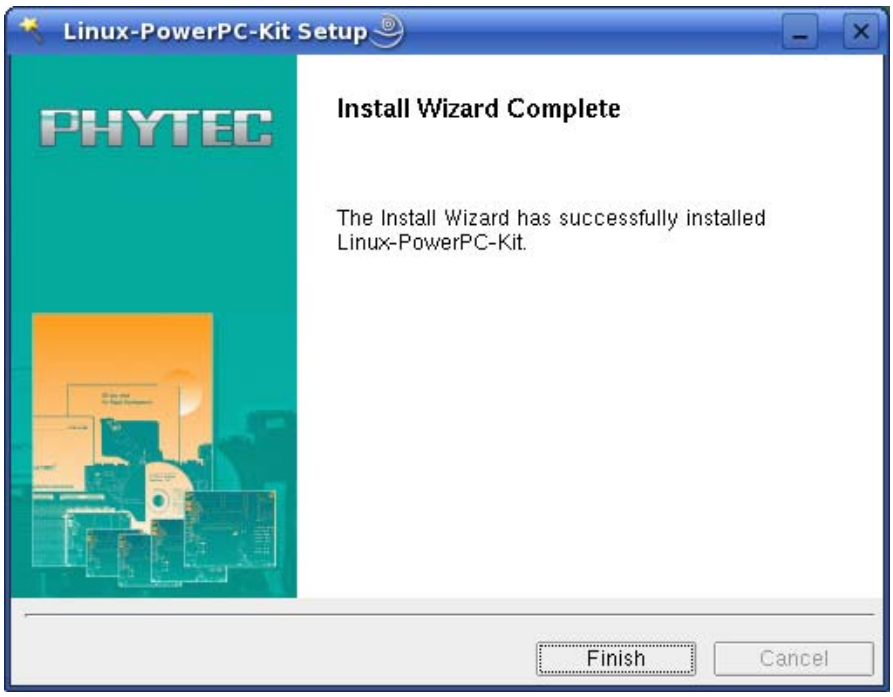
The GNU GCC C/C++ Toolchain will be installed to the standard default directory */opt/powerpc-603e-linux-gnu*. The program *mkimage* will be installed in */usr/local/bin*. All other programs and examples will be installed in the selected destination directory.

After the files are copied, a dialog box for the Eclipse installation will appear.

- Click on *Yes* to install Eclipse. If you want to skip the installation of Eclipse choose *No*.

We recommend installing Eclipse even if you have already installed Eclipse on your system. This version of Eclipse includes additional plugins.

- Click on *Next.*



- Click on *Finish* to exit the setup.
- Close the terminal window.

Now you will have to restart the KDE desktop.

- Open the *K Menu* from the lower-left corner of the desktop.
- Select the *Leave* tab and choose *Logout*
- When the display manager appears, enter your login name and password to restart the KDE desktop.

You have successfully installed the software for the Linux-PowerPC-Kit. You can find the programs you will need to develop own applications for the target on your host system. All necessary configurations were done by the setup program. In the following passage you can find some advanced configuration information.

Advanced Configuration Information

In this part you can find some information how to configure the configuration steps of the setup program by your own. All the following configuration steps are done by the setup program. The information in this part is for users who want to use the Linux-PowerPC-Kit on another Linux distribution than openSUSE. This is also interesting for users who want to see what configurations were done by the setup program.

During the setup program the GNU GCC C/C++ cross-compiler was installed in the directory */opt/powerpc-603e-linux-gnu*. To start the cross-compiler directly from every part of the system, the path of the cross-compiler was added to the environment path.

You can add the path of the cross-compiler to the environment by adding the following line in the file /etc/profile:

export PATH=/opt/powerpc-603e-linux-gnu/gcc-4.1.2-glibc-2.5-kernel-2.6.18/bin:$PATH

You can open a terminal program and use the cross-compiler directly from the command line. For example you can compile a C program with the following command:

powerpc-603e-linux-gnu-gcc HelloWorld.c –o HelloWorld

In the standard configuration only the user root has write access to the serial interface. To use a serial communication tool like Komport with normal user rights, you have to be a member of the group uucp.

A user can be added to this group with the following command:

groupmod -A <username> uucp

The serial communication program was configured during the setup with the following configuration:

115200 baud, 1 Start bit, 8 data bits, 1 stop bit, no parity and no flow control.

If you want to use another program than Komport for serial communication, you will have to setup this program with these settings.

2.4 Connecting the host with the target

In this section you will learn how to connect your host pc with the target. The connection will be done with a cross-over cable and serial cable. You will start Linux from flash on the target and you will be able to login with the serial communication program Komport and a telnet session via a peer-to-peer connection.

- Connect the serial cable with the lower connector P3 on the target and the serial interface COM1 on your host.

Ensure to use the cable included in this RDK.

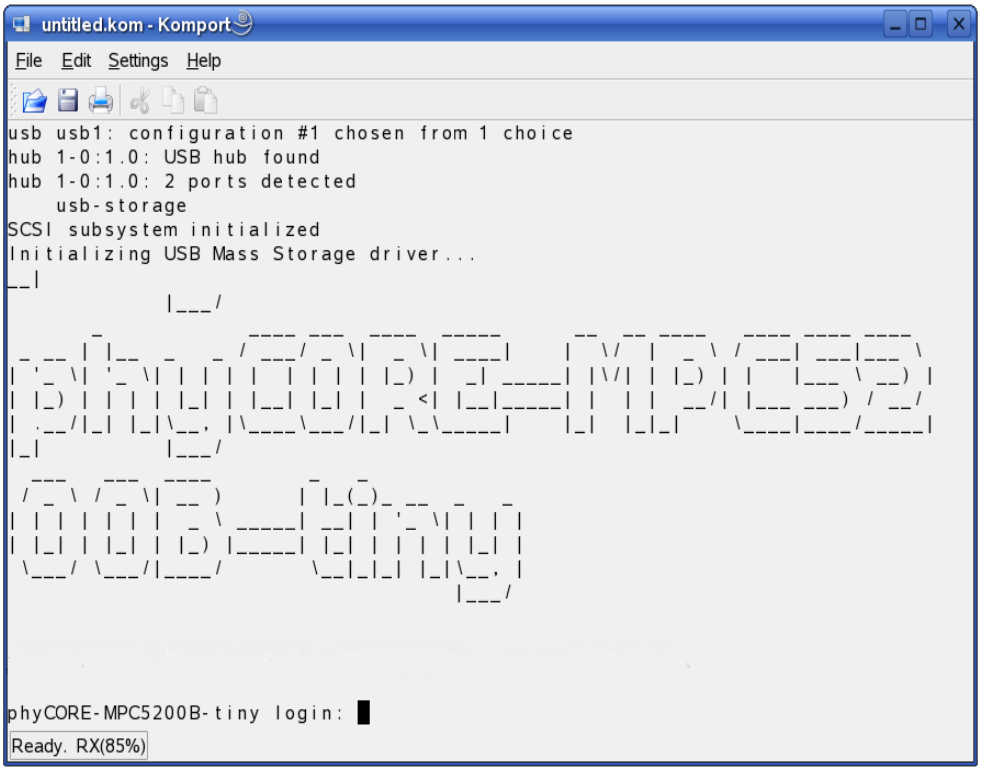
- Connect the cross-over Ethernet cable with the RJ45 connector of the target and the right network card of your host.

- Click on the *Komport* icon on your desktop.

- Connect the AC adapter with the power supply connector X6 (12V) on your board.

The power connector should have 12 VDC inside and outside should be ground.

After connecting the board with the power supply the target starts booting. When the target finished loading the file system you will see a screen similar to the following screenshot:



- Type **root** to login.
- After you have successfully logged in, you can close Komport.

When the target is connected with the Power supply first the bootloader U-Boot is loaded from the flash memory in the RAM. After the bootloader is uncompressing and booting the Kernel from the flash. When the Kernel is finished booting, the root file system is mounted. The used filesystem is JFFS2.

JFFS2 is a successor and a complete rewrite by Red Hat of the JFFS. As its name implies, the Journalling Flash File System Version 2 (JFFS2) implements a journalling filesystem on the memory technologie device (MTD) it manages. JFFS2 does not attempt to provide a translation layer that enables the use of a

traditional filesystem with the device. Instead, it implements a log-structured filesystem directly on the MTD device. The filesystem structure itself is recreated in RAM at mount time by JFFS2 through a scan of the MTD device's log content.

In addition to its log-structured filesystem, JFFS2 implements wear leveling and data compression on the MTD device it manages, while providing power down reliability. .JFFS2 can gracefully restart and is be capable of restoring a filesystem's content without requiring outside intervention regardless of power failures.

Troubleshooting:

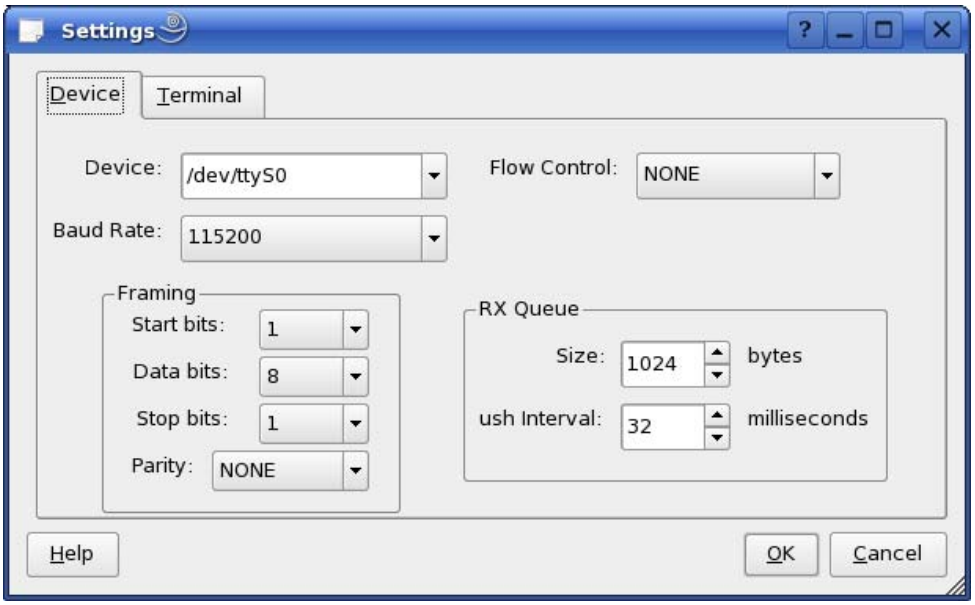
If you do not see an output in the Komport window, check the connections between the target and your host.

At the end of the setup you had to restart the KDE desktop. If you haven't done yet, restart the KDE desktop and try again.

Another problem could be a wrong configuration of Komport.

- Choose in the menubar of Komport *Settings –> Configure Komport.*

Komport should be configured with the configuration in the following dialog:



- If you don't have these settings, adapt your settings accordant to the screenshot.

One problem could also be a missing read / write access to the serial interface:

- Open the *YaST Control Center*.
- Choose *Security and Users.*
- Choose *User Management*

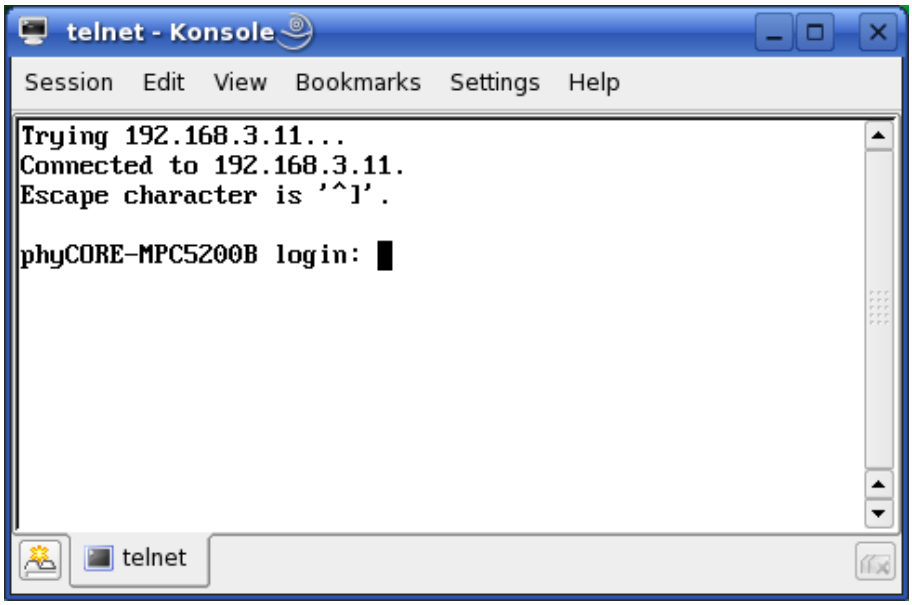
In the line of you user name should be the group uucp.

- If the group is missing select your user name and click on *Edit*
- Select the tab *Details*.
- Check in *Groups* the group *uucp*.
- Click on button accept.
- Click on Finish and close YaST.

Now you can test the network connection to the target.

- Click on the Icon *Telnet for Target* on the desktop.

A new window with a connection to the target opens.



If you see the user login in the opened window, the network configurations were configured correctly.

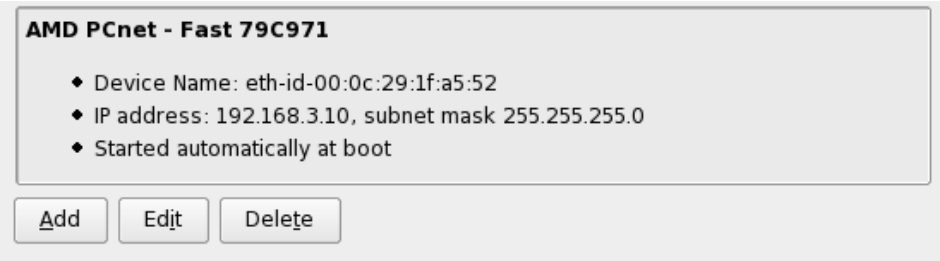
- Close the window.

You have successfully setup all configurations to access your target from your host.

Troubleshooting:

If you don’t see the user login, check the connection between the target and the host. If you have installed more than one network card on your host, be sure to connect the cable with the network card you have configured with the ip address 192.168.3.10.

If you don’t see the login, you may not have set up the right ip address of your host. You can check the settings of your network card by opening YaST. In the YaSt Control Center you can select *Network Card* in *Network Devices.* There should be the following configuration:



Information how to configure your network device can be found in the section *Configuring the Host Plattform.*

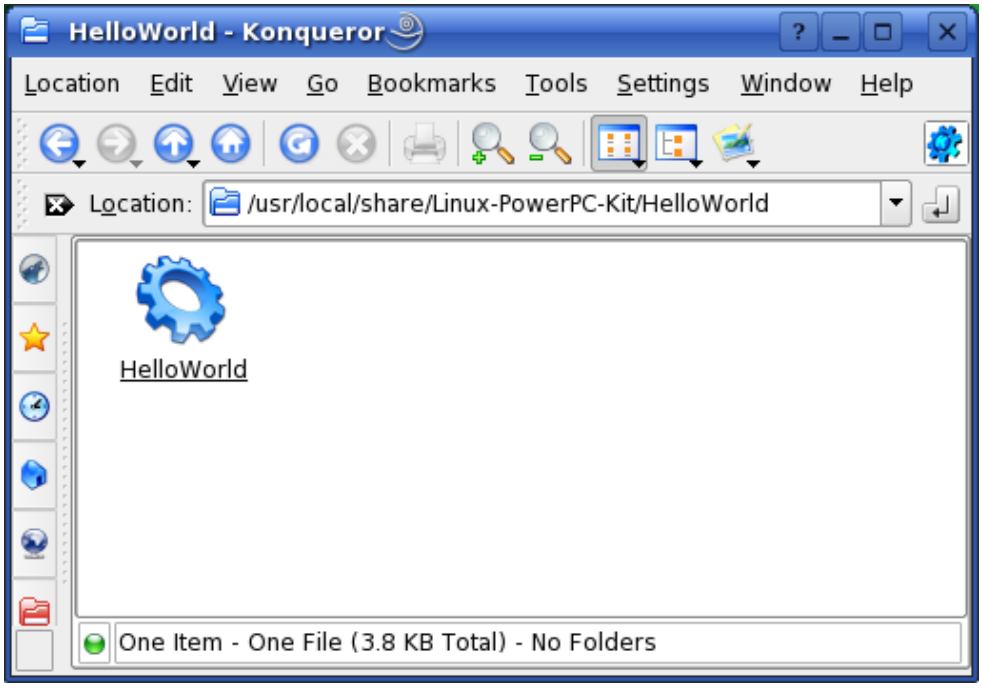
2.5 Copying an Example to the Target

In this section you will learn how to copy an example program to the target using the FTP protocol with the Konqueror browser. After that you will execute the example on the target. At the end of this passage you can find some information how to copy and execute a file on the target using the command line.

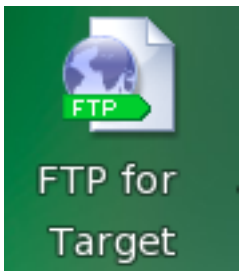
Copying a program to the target:

- First click on the icon *Linux-PowerPC-Kit* for KPCM-030 or *Linux-PowerPCIO-Kit* for KPCM-032on the KDE desktop.

A new window with the content of the installation directory opens.

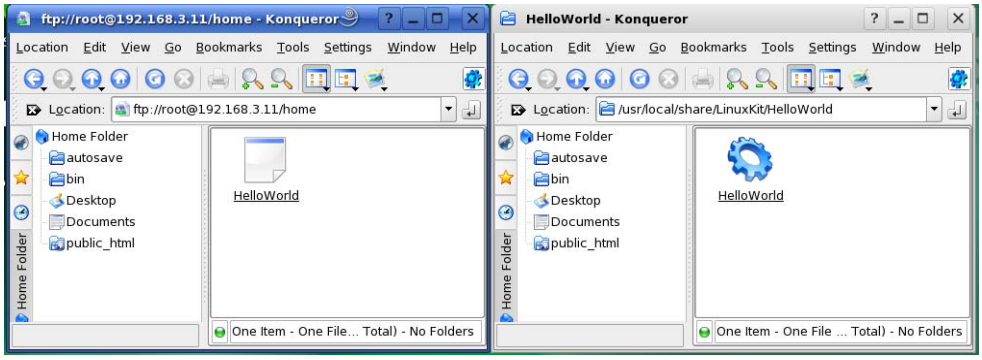


- Click on the directory *HelloWorld*.



- Click on the icon *FTP for Target* on the desktop.

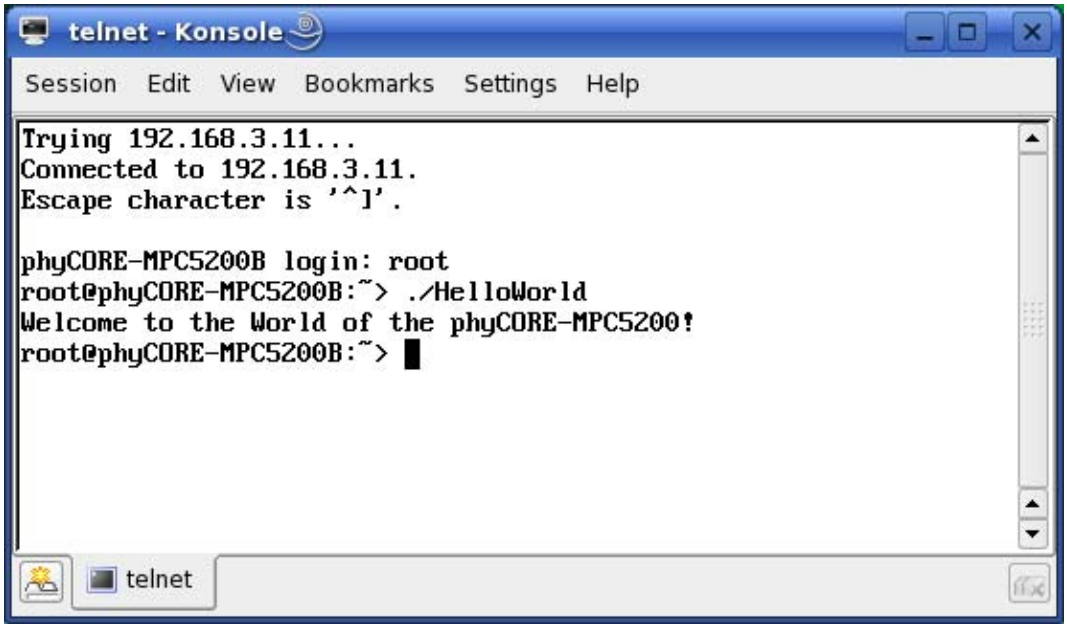
A window with a FTP session to the target opens. Now you have two windows, one for the target and one for the host. You can use these two windows to copy files per drag and drop from the host to the target.



- Select the window with the *HelloWorld* program.
- Click on the *HelloWorld* program and hold the left mouse button pressed.
- Pull the program into the window with the FTP –session to the target and release the mouse button.
- Choose *Copy here* in the appearing selection menu.
- Close these two windows.

Executing a program on the target:

- Click on the Icon *Telnet for Target* on the desktop.



- Enter **root** as login name and press **Enter**.
- Enter **./HelloWorld** and press **Enter.**

The program starts and you see the following output:

Welcome to the World of the phyCORE-MPC5200

Using SSH to execute a program on the target:

SSH can be used if you want to execute a program directly from the host on the target. Later this will be used to execute programs out of Eclipse on the target. To start programs out of Eclipse you have to log into target with SSH from the command line for one time. This is necessary to add the RSA public key of the target to the list of the known hosts.

There are several authentication methods to use ssh. The used method is the *hosts.equiv* method combined with RSA-based host authentication.

If the machine the user logs in from is listed in /etc/hosts.equiv on the remote machine, and the user names are the same on both sides, the user is considered for log in.

On the target the file /etc/hosts.equiv has the following entry:

```
# file: /etc/hosts.equiv
#
# Allow access from everywhere.
#
+ +
```

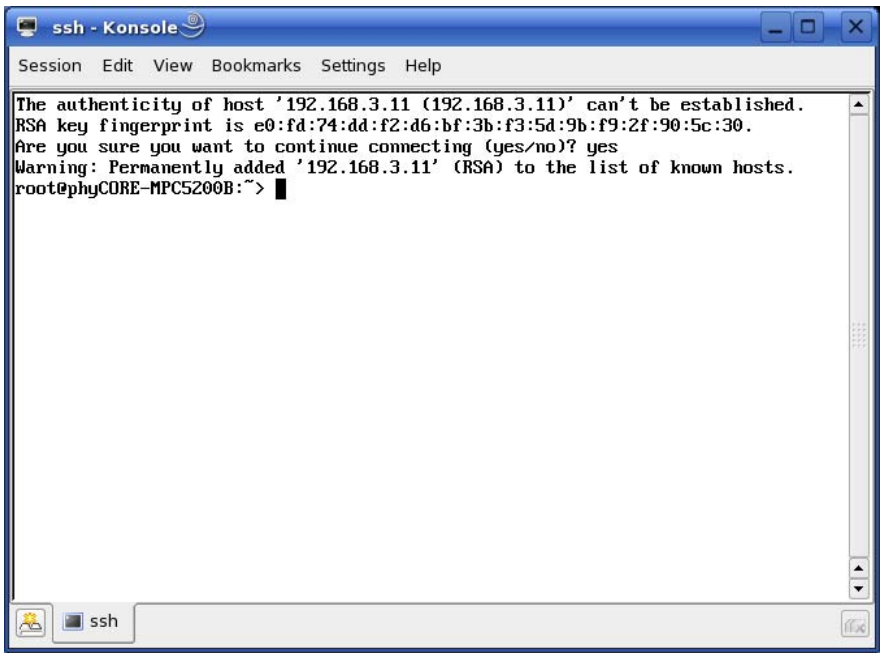
The "+ +" means that every user can login from everywhere.

When the client is connected to the target the file $HOME/.ssh/known_hosts is consulted when using hosts.equiv with RSA host authentication to check the public key of the target. The key must be listed in this file to be accepted. The client uses the same file to verify that it is connecting to the correct remote host. Then the client is connected to the host for the first time, the RSA public key of the host can be added to the file $HOME/.ssh/known_hosts automatically. Then the client can be connected to the host without entering a password.

This authentication method closes security holes due to IP spoofing, DNS spoofing and routing spoofing. Note /etc/hosts.equiv in general, is inherently insecure and should be disabled if security is desired.

- Click on the Icon *SSH for Target* on the desktop.

A new window opens.



In this window you can see that the authenticity of the host can't be established. This is normal if you want to create a connection for the first time.

- Enter **yes** and press **Enter** to continue.
  The RSA public key of the target will be permanently added to the list of the known hosts.

Troubleshooting:

If an error occurs and you can't see the *root@phyCORE-MPC5200B:~>* prompt open a terminal window and enter the following command:

- **rm ~/.ssh/known_hosts.**

Try to login and enter:

- **ssh root@192.168.3.11**
- Enter **yes** to add the target to the known_hosts.

Now you should see the login prompt.

We expect that you haven't changed the ssh configuration file on your host. If you change this file the authentication may not work.

Now you are logged in and you can execute programs on the target.

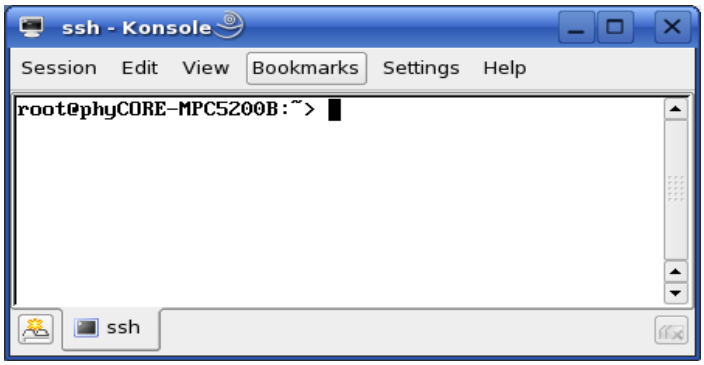
- Type **./HelloWorld** start the program.

The program starts and you see the following output:

Welcome to the World of the phyCORE-MPC5200!

- Close the ssh window.

If you click on the Icon *SSH for Target* again, the host will be connected directly to target and you will see the following screen:



- Close the window.

You have successfully copied and executed an example on the target.

Advanced Information

Copying a program to the target with the command line:

- Open a new terminal window.
- Change to */usr/local/share/Linux-PowerPC[IO]-Kit/HelloWorld*:

  cd /usr/local/share/Linux-PowerPC[IO]-Kit/HelloWorld
- Create a FTP-Session to the target: **ftp root@192.168.3.11**
- Press *Enter* (no password required).
- Copy the application file to the target by typing:

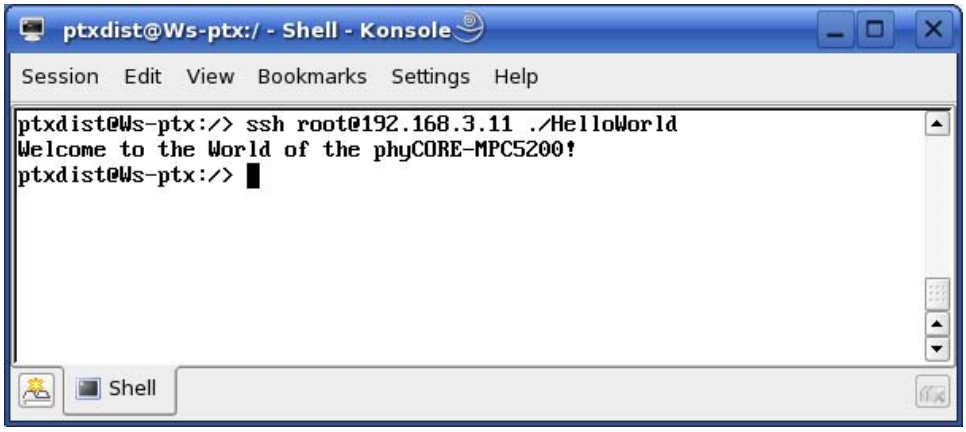
ftp>put HelloWorld

- End the ftp session: **ftp>exit**

Executing a program on the target:

- Open a telnet session to the target: **telnet -l root 192.168.3.11**
- Type **root** and press *Enter*.
- Type **./HelloWorld** to start the application.

Executing a program on the target using ssh:



- Type **ssh root@192.168.3.11 ./HelloWorld to start the program.**

After the program is executed ssh will logout automatically.

3 Getting More Involved

70 min

In this chapter you will pass some continuative topics. First you will configure and compile your own kernel. With the kernel configuration tool you can add additional features or disable them, if they are not needed. After compiling the kernel, you will learn how to write the newly created kernel into Flash and start the target with the new kernel.
Then you will start working with the Eclipse Platform using the C/C++ Development Tools (CDT) in conjunction with the GNU GCC C/C++ Toolchain. You will learn how to configure the Eclipse Platform and open an existing project. After that you will create your first own project and modify the source code from the example.
At the end of this chapter you will execute the program as external applications out of Eclipse and add your created application to the startup configuration of the target.

3.1 Configuring and Compiling the Kernel

In this part you will learn how to configure and build a new kernel. First you will copy the kernel archive to your home directory and extract the kernel source. Then you will configure the kernel with the graphical user interface of qconf. After the configuration you will compile the new kernel using the GNU cross development toolchain.

The used kernel is a standard kernel you can find on the homepage of www.kernel.org. The kernel archive in your setup installation directory already includes the patches for the phyCORE-MPC5200.

At the first step open a new terminal.

- Click on the terminal icon on your desktop.
- Type the following commands to copy the kernel archive to your home directory:

  cp /usr/local/share/Linux-PowerPC[IO]-Kit/linux-2.6.x.tar.bz2 ~

  cd ~

- Unpack the kernel source archive:

  tar –xvjf linux-2.6.x.tar.bz2

- Change to newly created directory:

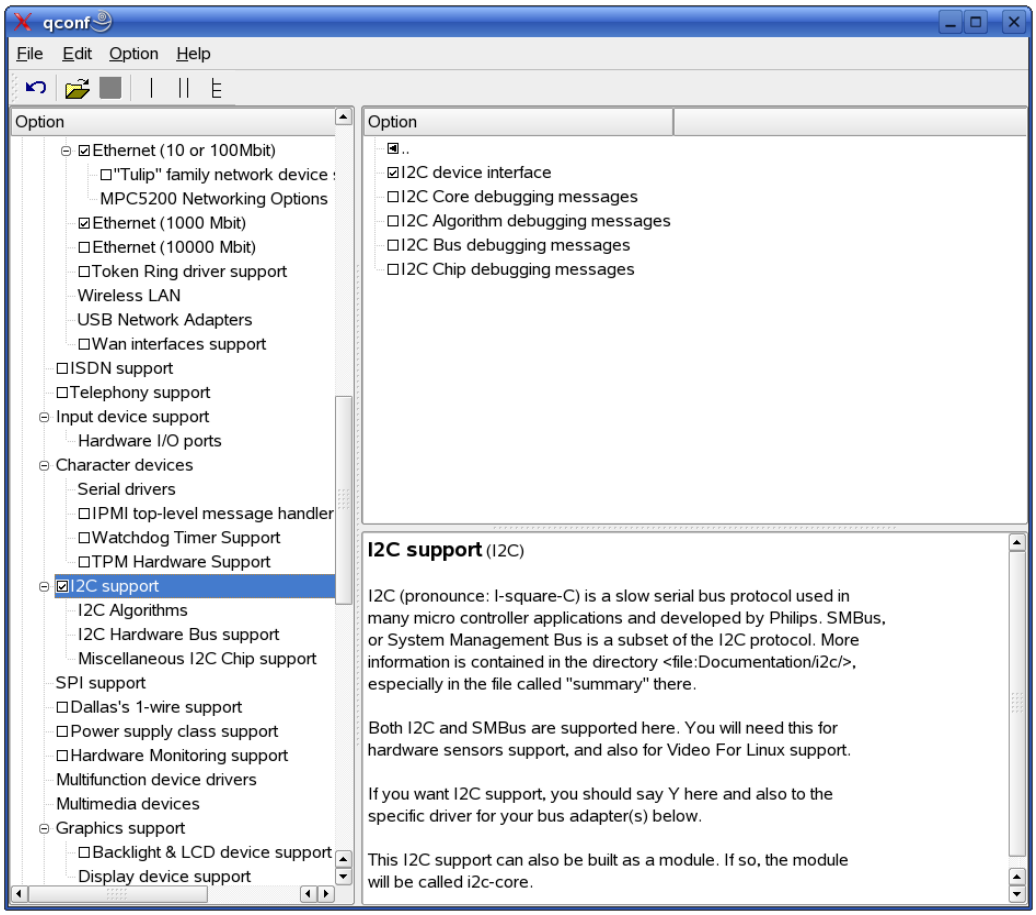
  cd linux-2.6.x

The kernel usually builds the system for the native machine architecture. To use the powerpc architecture and powerpc cross-compiler, you will have to specify the architecture and the cross-compiler in the command line.

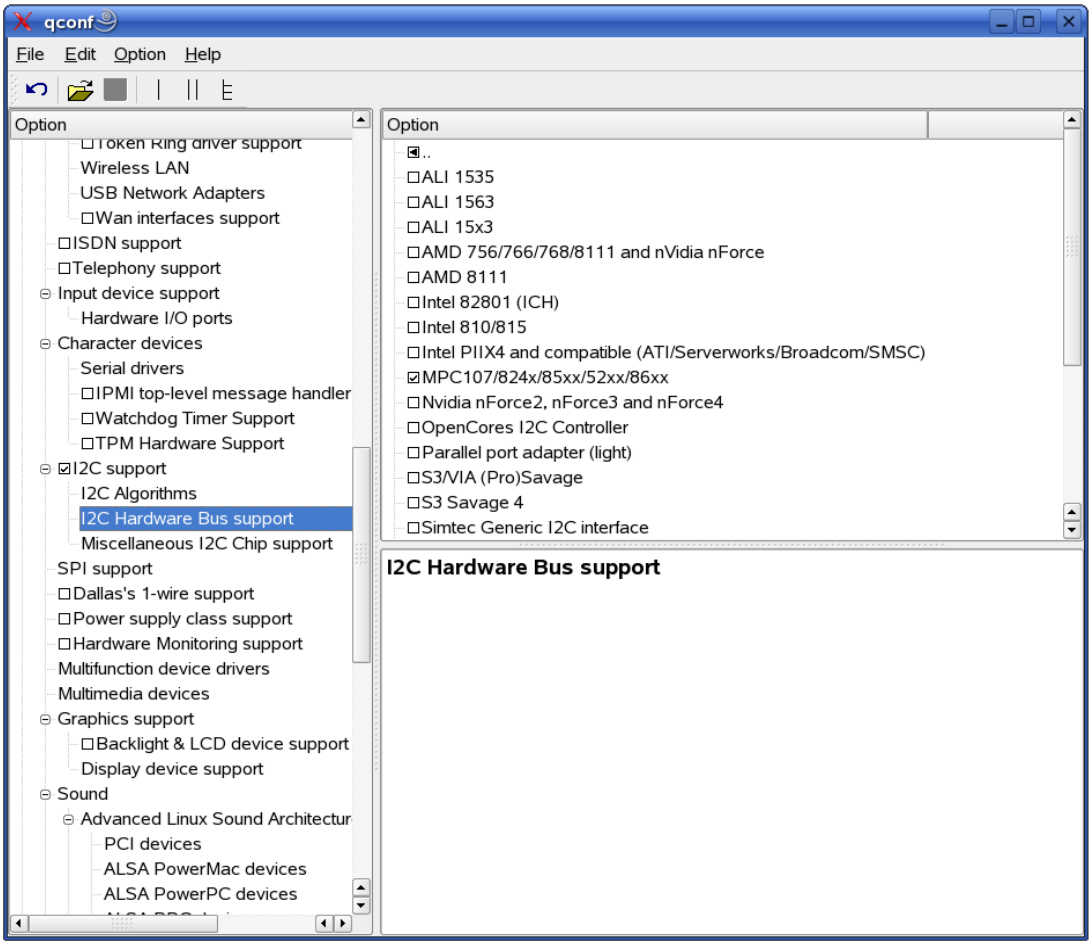
- Type:

  make xconfig ARCH=powerpc

The kernel configuration tool *qconf* starts.



- Check *I2C support* in *device drivers.*
- Check *I2C device interface.*



- Click on *I2C Hardware Bus Support.*
- Check *MPC107/824x/85xx/5200.*
- Save your configuration and exit the configuration tool.
- Type the following command (one line):

  make ARCH=powerpc

  CROSS_COMPILE=powerpc-603e-linux-gnu- uImage

The kernel sources will be compiled and the new kernel will be built. The time of building the new kernel depends on your host platform. You will find the new kernel **uImage** in *arch/powerpc/boot.*

If the process of building the kernel stops with an error, check the parameter of ARCH and CROSS_COMPILE.

- Close the terminal window.

In this section you learned how to configure and compile a new kernel. Now you are able add new features to the kernel.

3.2 Writing the Kernel into Flash

In this passage you will find a description how to write the newly created kernel into the flash memory. Before the kernel can be written into flash, you will have to download the kernel from a tftp server. This will be done in the command line of the bootloader. The kernel will be copied into RAM, starting at address 0x400000. Then you will have to erase the part of the FLASH, where you want to copy the Kernel image.

In the default configuration you will find five partitions on the target. The first partition contains a second bootloader for rescue issues, the second partition contains the kernel, the third the root file system, the forth contains the bootloader and the last contains some free space at the end of the FLASH.

The five partitions have the following address ranges for KPCM-030:

0xFF000000 - 0xFF03FFFF U-Boot-Low (256k)
0xFF040000 - 0xFF1FFFFF Kernel (1792k)
0xFF200000 - 0xFFEFFFFF Root-FS(13312k)
0xFFF00000 - 0xFFF3FFFF U-Boot (256k)
0xFFF40000 – 0xFFF7FFFF Oftree (256k)
0xFFF80000 - 0xFFFFFFFF Space (768k)

The five partitions have the following address ranges for KPCM-032:

0xFE000000 - 0xFE03FFFF U-Boot-Low (256k)
0xFE040000 - 0xFE1FFFFF Kernel (1572k)
0xFE200000 - 0xFFEFFFFF Root-FS(30408k)
0xFFF00000 - 0xFFF3FFFF U-Boot (256k)
0xFFF40000 - 0xFFFFFFFF Space (768k)

You should never erase the U-Boot partition. If this partition is erased, you can't restart your target.

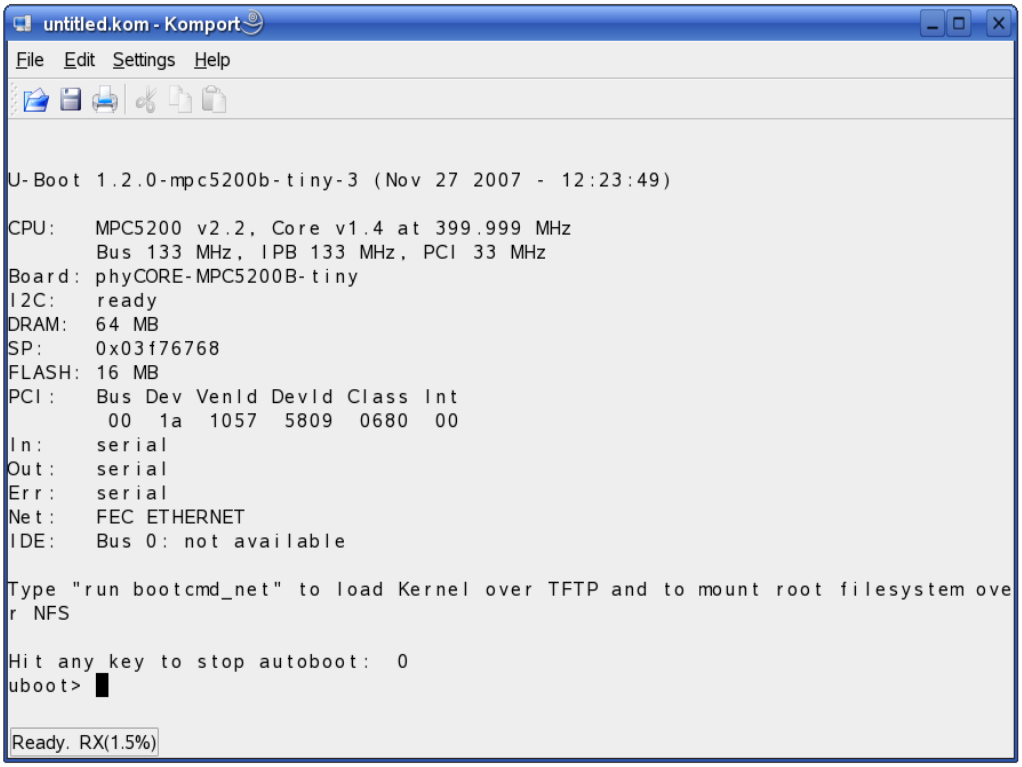
- First open a new terminal window, if is not open yet.
- Change to the kernel directory:

  cd ~/linux-2.6.x/arch/powerpc/boot
- Change to user root:

  su root
- Copy new Kernel image to the directory tftpboot :

  cp uImage /tftpboot

  exit
- Close the terminal window.



- Open Komport and press the RESET button on the target.

You will see the output *Hit any key to stop autoboot.*

- Press any key to stop autoboot.

You can download the kernel from the TFTP – server, erasing the required Flash area and writing the kernel from the RAM into the Flash with one simple command.

Before you can execute this command you have to set the name of your kernel image to the environment variable *uimage*. Then you can start downloading and writing the kernel image into the Flash using the command *run prg_kernel*.

```
Hit any key to stop autoboot:  0
uboot> setenv uimage uImage
uboot> run prg_kernel
Using FEC ETHERNET device
TFTP from server 192.168.3.10; our IP address is 192.168.3.11
Filename 'uImage'.
Load address: 0x400000
Loading: #################################################
  #################################################################
  #################################################################
  ###################
done
Bytes transferred = 1093419 (10af2b hex)

.............. done
Erased 14 sectors
Copy to Flash... done
uboot>
```

- Type the following command to set the name of your kernel image:
  setenv uimage uImage

- Type **run prg_kernel** to download and write the kernel into the Flash

The copy process can take up to a minute, depending on the speed of your system.

- Press the RESET button on the target to restart with the new kernel.

The target will restart booting with the newly created kernel.

- Close Komport when the target successfully finished with booting the kernel and mounting the root filesystem.

Troubleshooting:

If any problem occurs after writing the kernel into Flash, you can write the kernel and root-file system from you setup CD into the Flash.

You can find the kernel and root file system in the directory Linux\image.

- To restore your system for KPCM-030 copy the files *uImage-pcm030-x.0* and *root-pcm030-x.0.jffs2* into the directory */tftpboot*
  Type the following commands in the u-boot command line
  setenv uimage uImage-pcm030-x.0
  setenv jffs2 root-pcm030-x.0.jffs2
  run prg_kernel
  run prg_jffs2

- To restore your system for KPCM-032 copy the files *uImage-pcm032-x* and *root-pcm030-x.jffs2* into the directory */tftpboot*
  Type the following commands in the u-boot command line
  setenv uimage uImage-pcm032-x
  setenv jffs2 root-pcm032-x.jffs2
  run prg_kernel
  run prg_jffs2

In this section you learned how to download a kernel image from a tftp server into the RAM memory of the target. The kernel was written from RAM to flash and the target was started with the new kernel

.

3.3 Opening an Existing Project

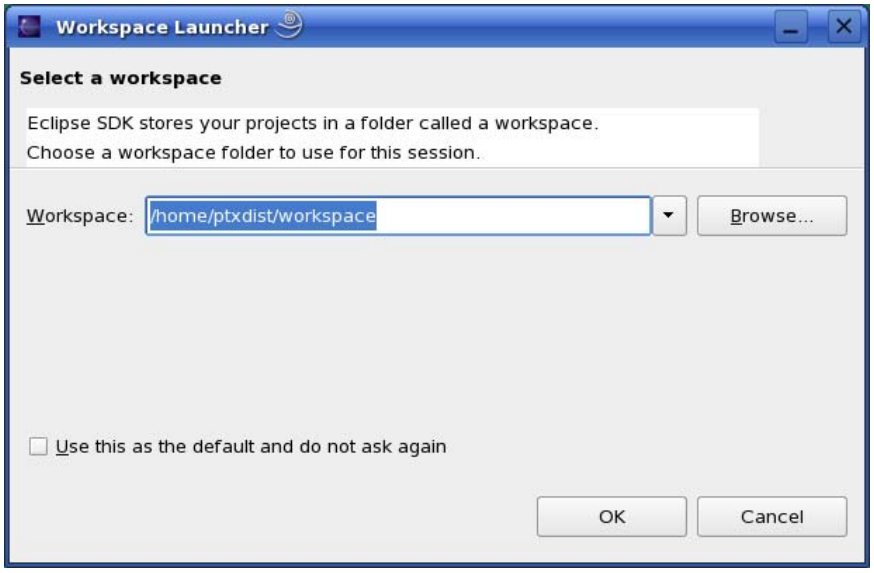
In this section you will import an existing Eclipse project into your workspace. The imported example project will be compiled with the cross-compiler. After compiling the project you will copy and execute the newly created program on the target.

Copying the HelloWorld project:

- Click on the *Linux-PowerPC[IO]-Kit* directory icon.
- Click on *examples_eclipse*
- Right-click the *HelloWorld* directory and select *Copy*.
- Browse to the directory */home/<your Home>*
- If the *workspace* directory doesn't exist, create a directory *workspace* in your home directory.
- Select the *workspace* directory.
- Right-click in the *workspace* directory and select *Paste*.
- Close the *Konqueror* file browser.

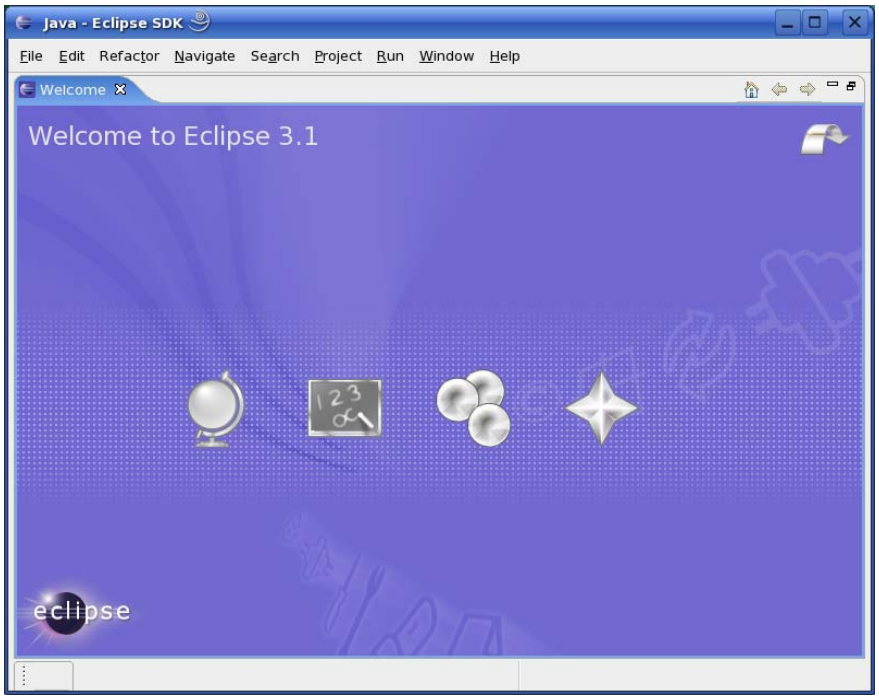
Starting Eclipse and importing the example project:

- Click on the *Eclipse* icon to start the application. You can find the icon on your desktop.

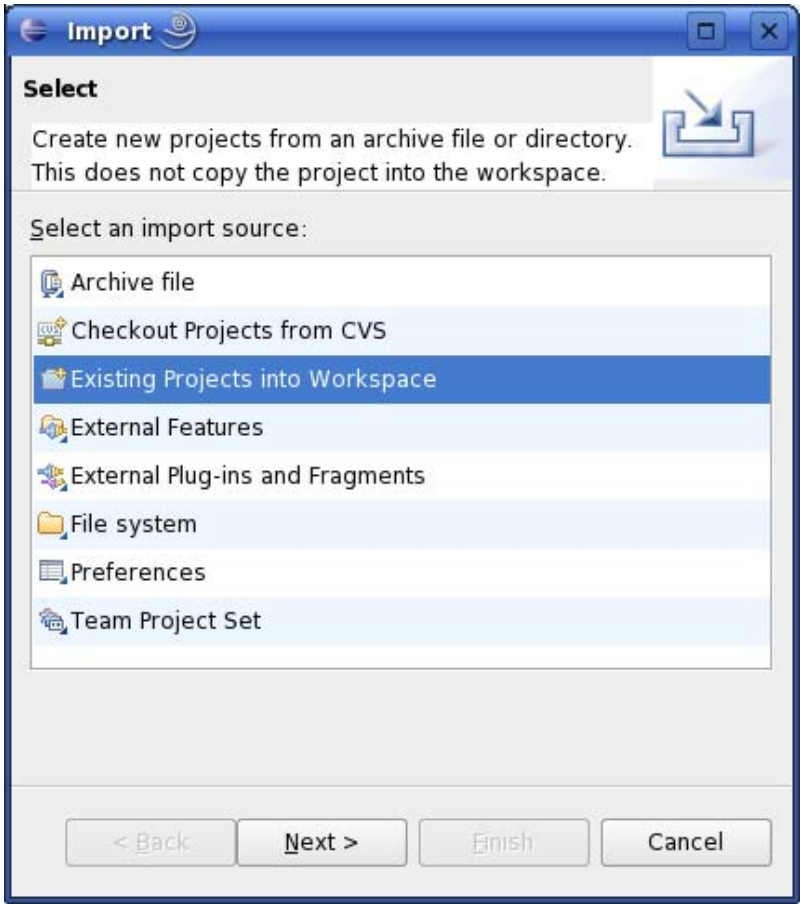


- Confirm the workspace directory with *OK*.

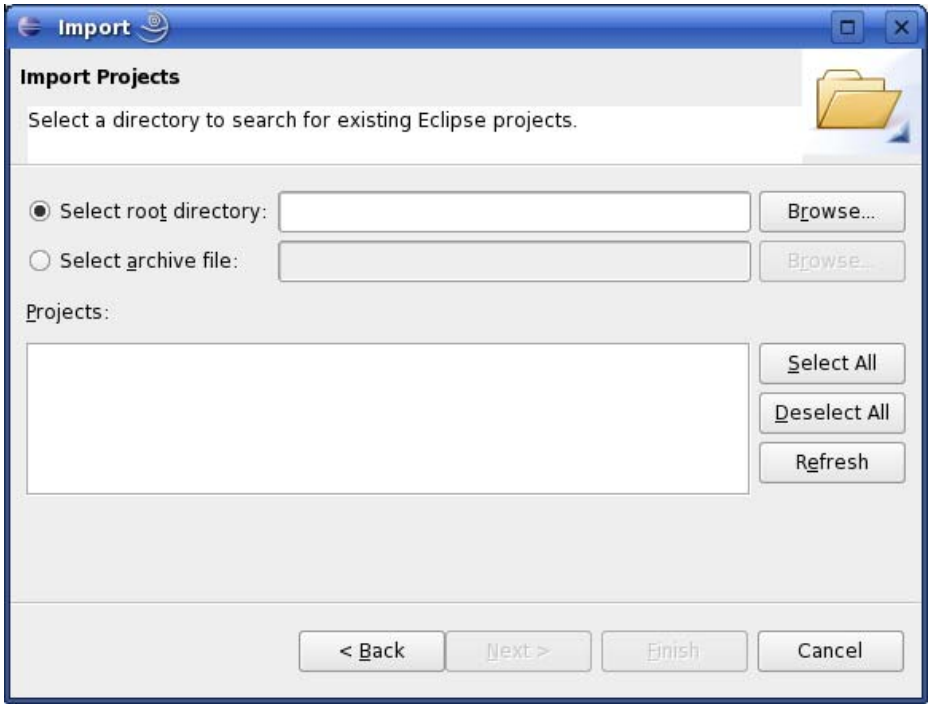
The *Welcome* screen will appear.



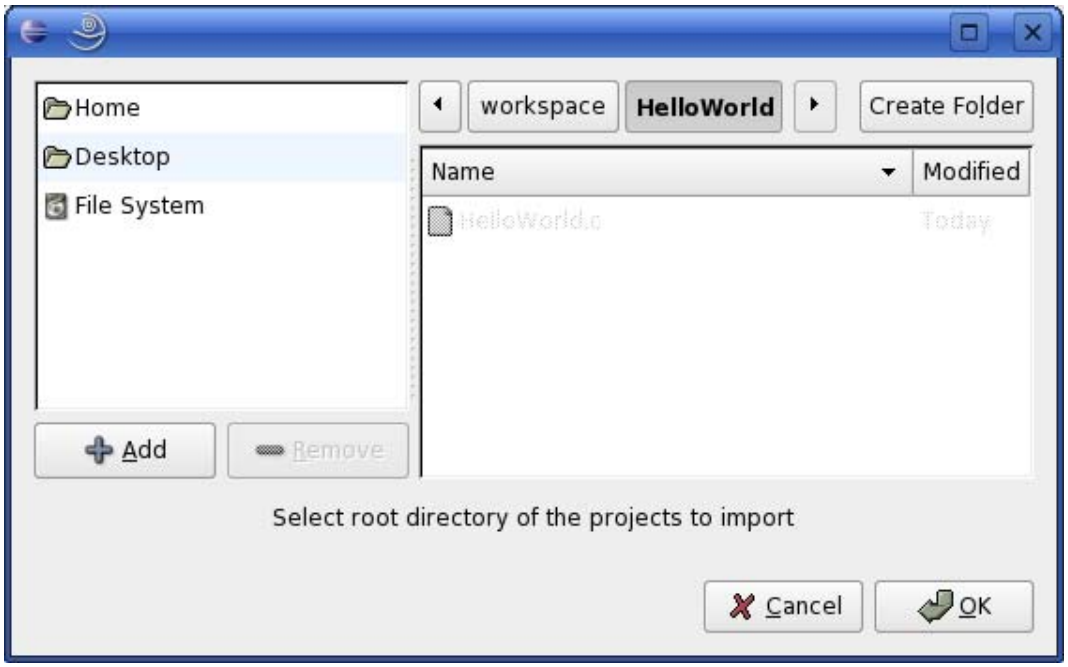
- Select *File* -> *Import* in the menu bar.



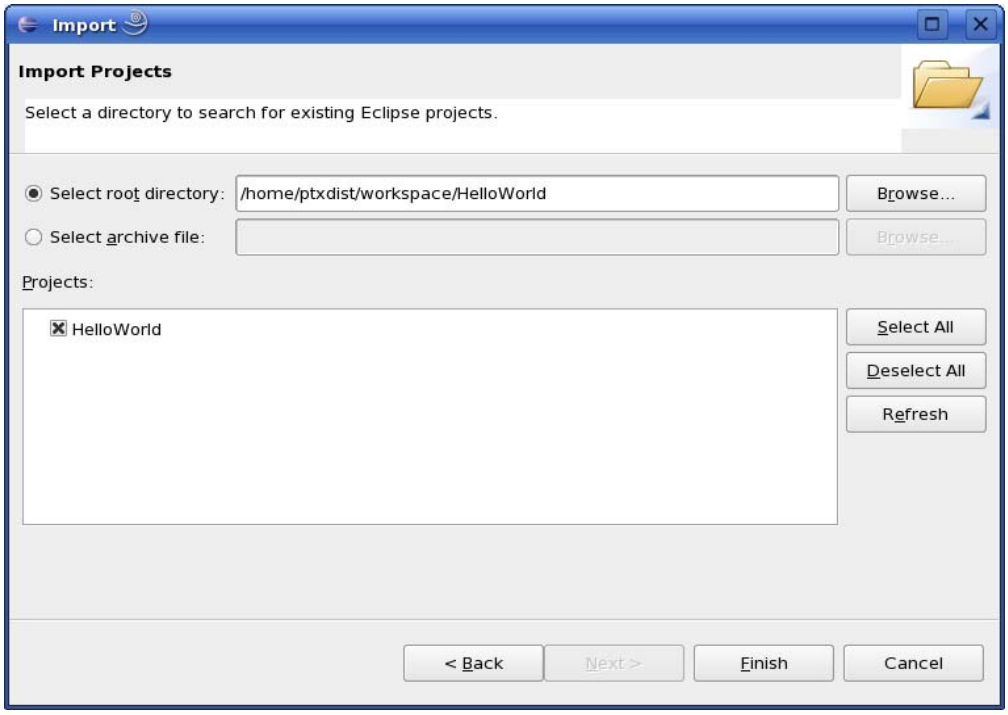
- Select *Existing Projects into Workspace.*
- Click on button *Next*.



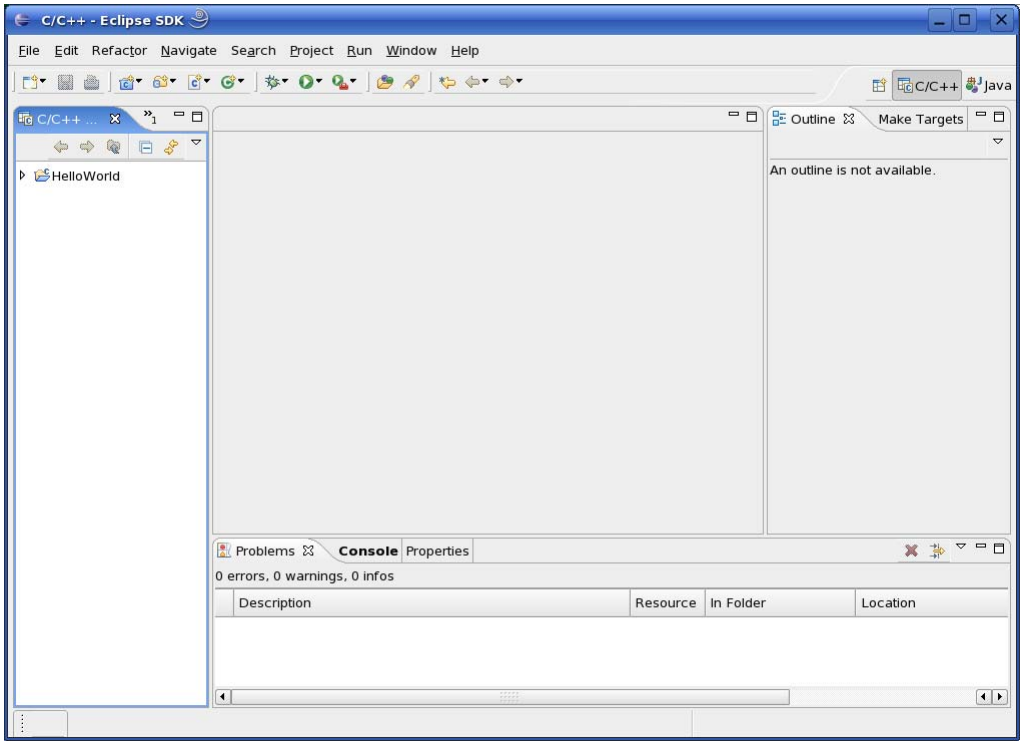
- Select *Browse.*



- *Double-Click* on the *HelloWorld* directory.
- Click on the *OK* button.



- Select *Finish* to import the project.

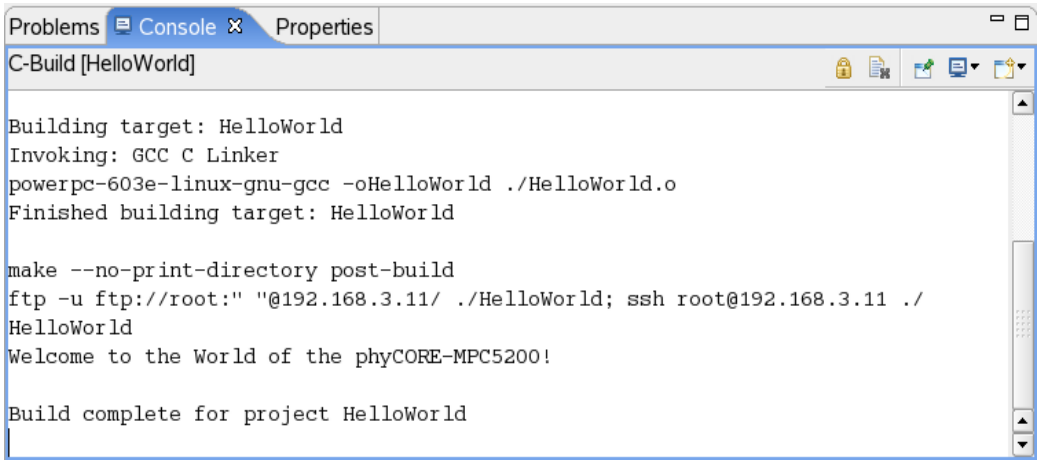


- Close the *Welcome Screen*.

The *HelloWorld* program will be compiled and the HelloWorld executable is built for the target. The HelloWorld file is also copied to the target using the ftp command. When the file is copied to target, the program is executed on the target using ssh. You can see the "Welcome to the .. " output in the console window.

- Select the *Console* tab.

You will see the following content in the console window:



If the project is not built automatically, you will have to check in the menu bar *Project -> Build automatically.*

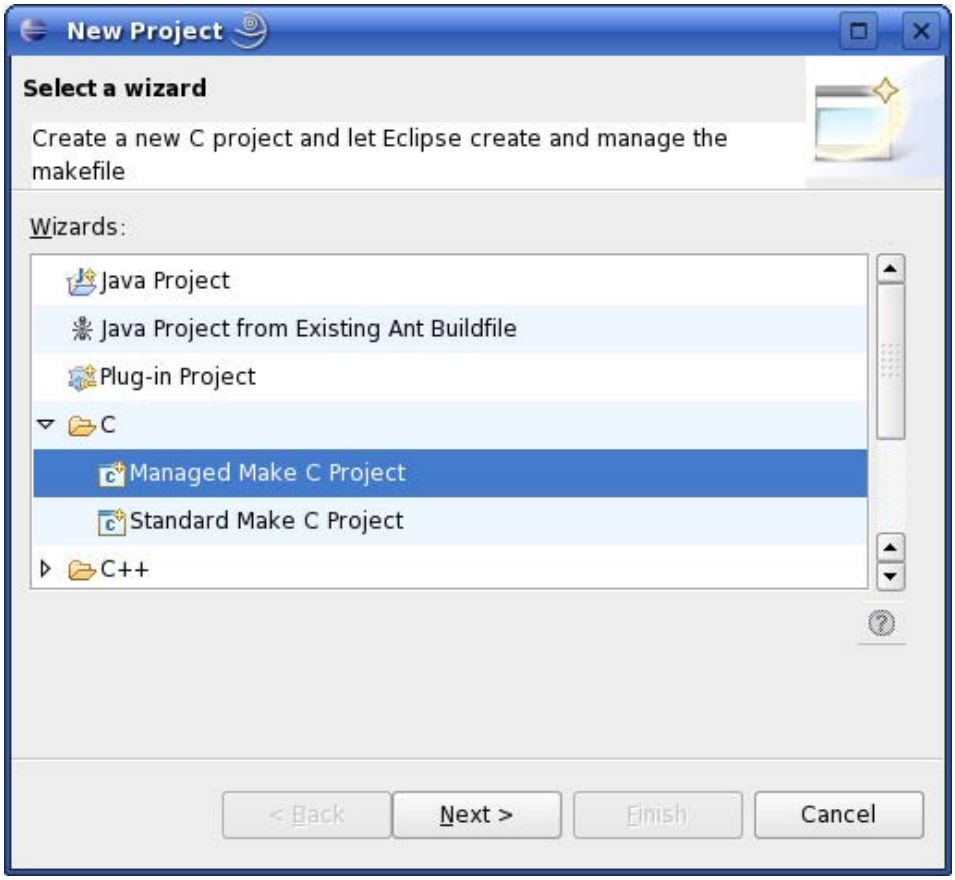
You have successfully passed the first steps with the Eclipse IDE. You are now able to import existing projects into the eclipse workspace. You can compile an existing project and execute the program on the target.

3.4 Creating a New Project

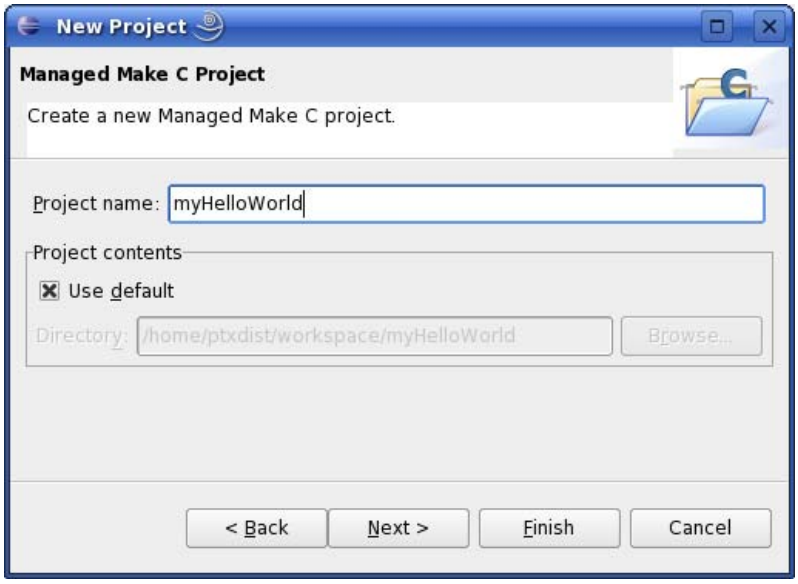
In this section you will learn how to create a new project with Eclipse and how to configure the project to use the GNU C/C++ Cross Development Toolchain.

- Open Eclipse if it isn´t already open.
- Select in the menu bar *File -> New -> Project.*

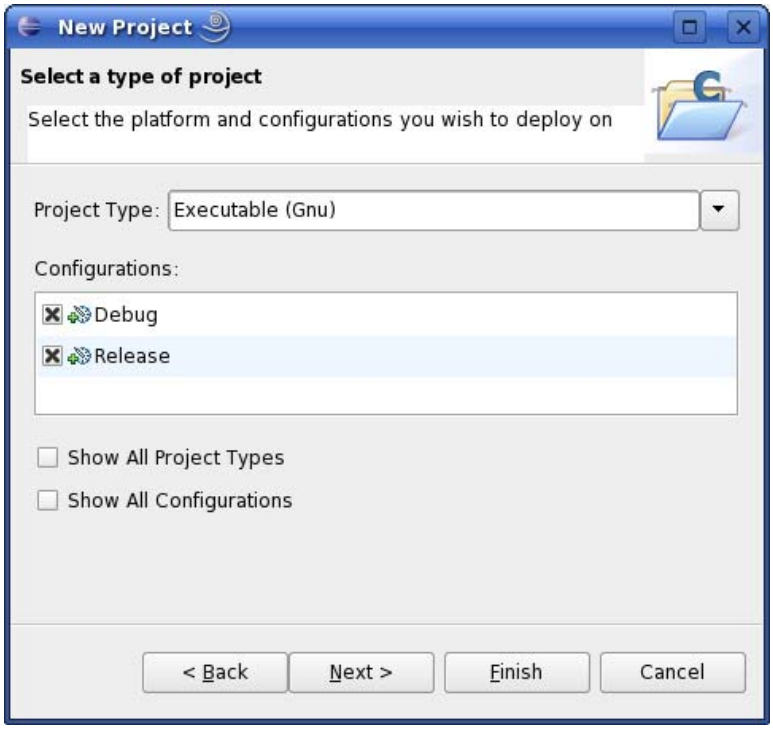
A new dialog opens.



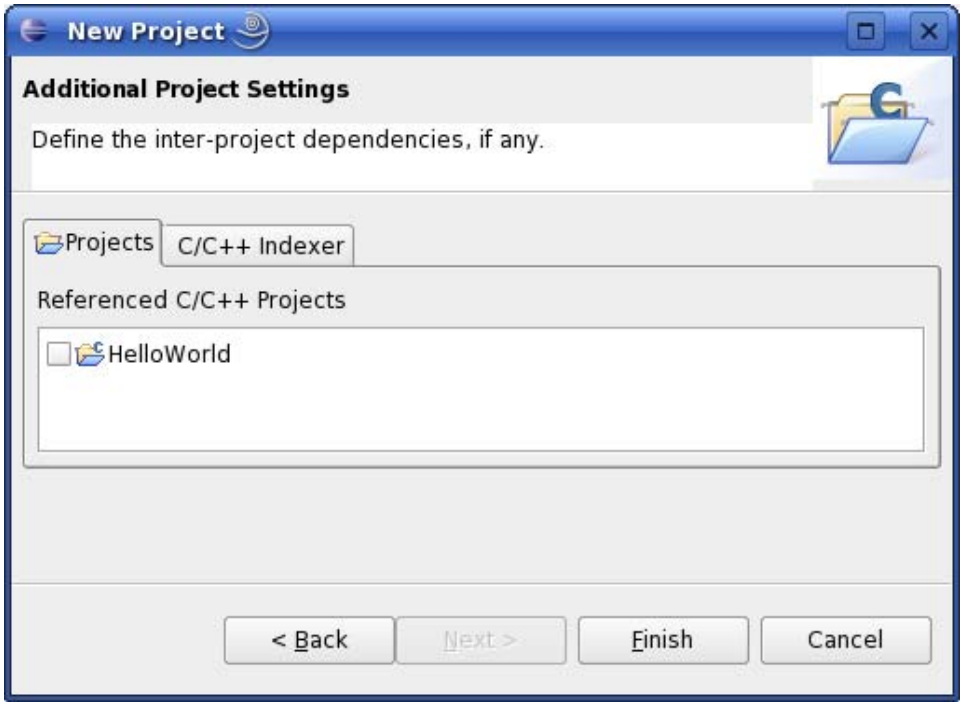
- Select *Managed Make C Project* and click on *Next*.



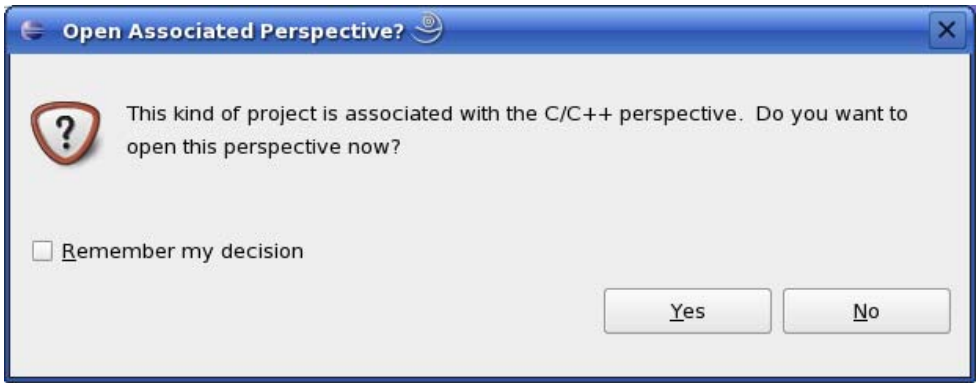
- Enter the project name *myHelloWorld* and click on *Next*.



- Click on button *Next*.

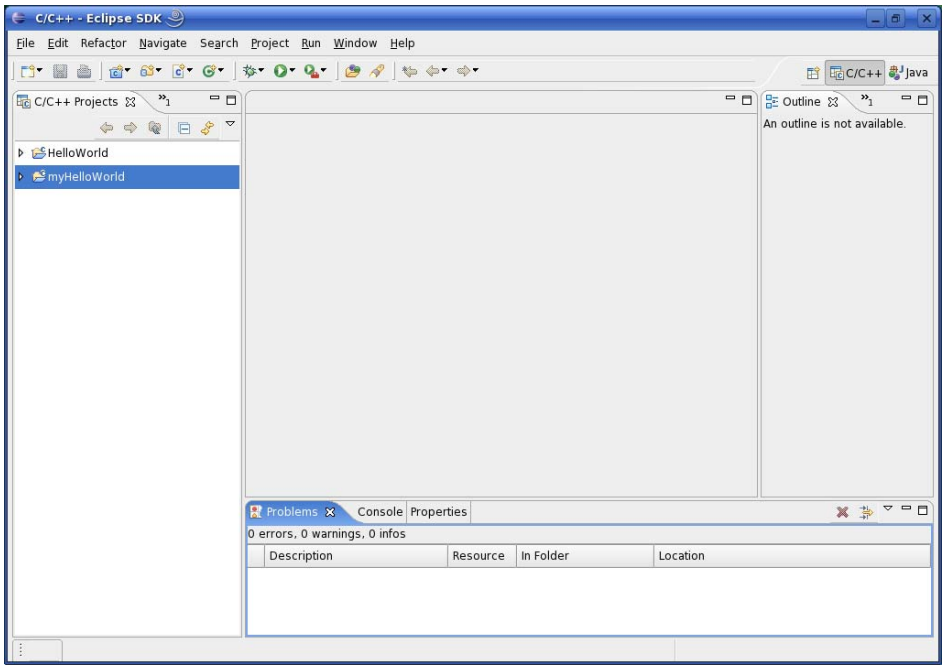


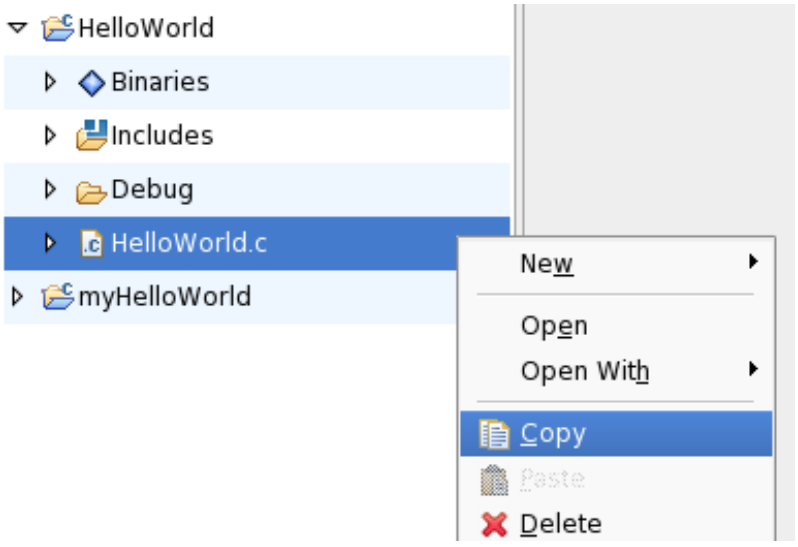
- Click on button Finish.



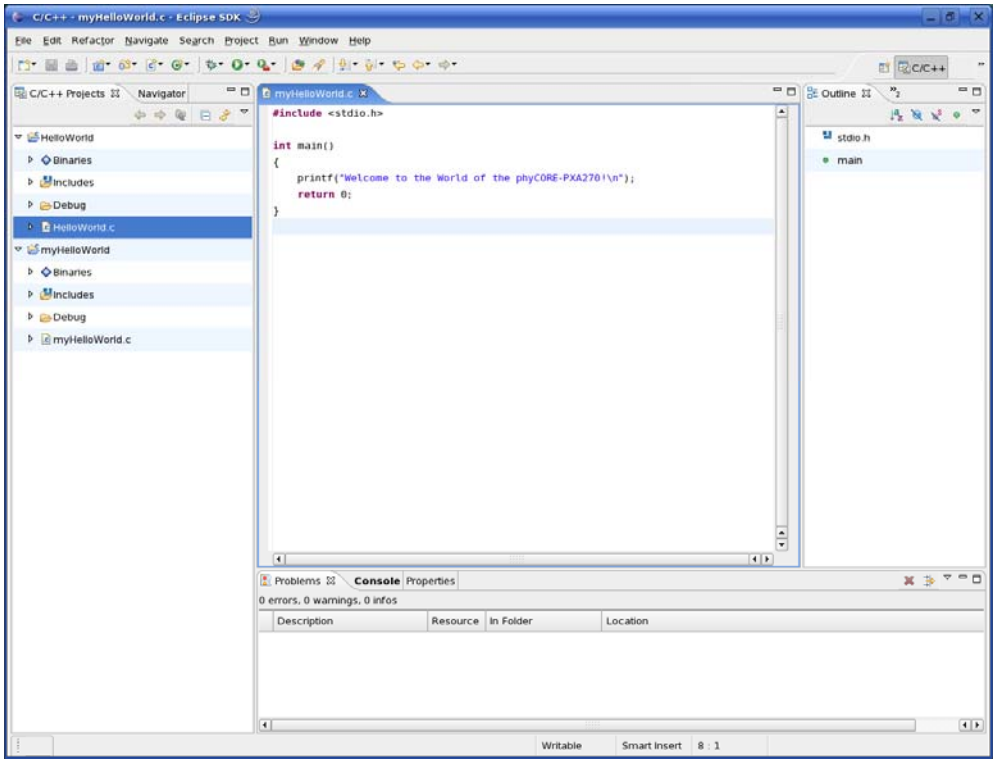
- Select *Yes* to open the C/C++ perspective.

You will see the C / C++ IDE with the created Project *myHelloWorld*.





- Right-Click on *HelloWorld.c* in the *HelloWorld* Project.
- Select *Copy*.



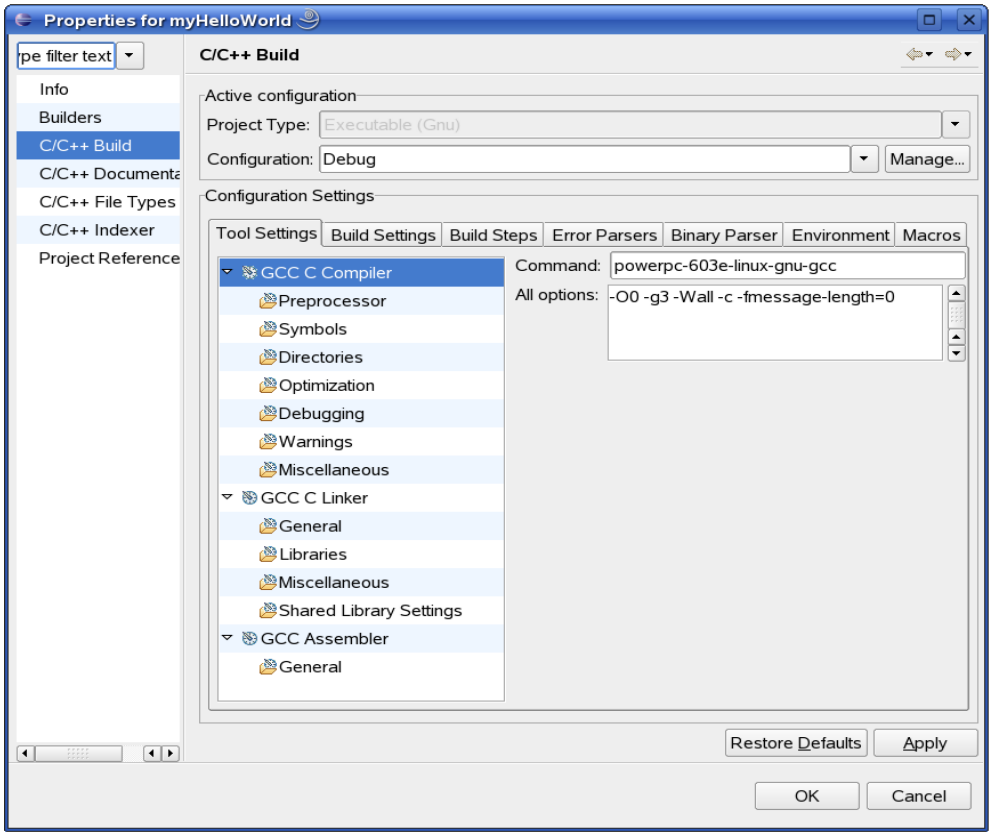
- Select the *myHelloWorld* project.
- Right-Click and select *Paste*.
- Double-Click on *HelloWorld.c* in the project *myHelloWorld*.

If *Build Automatically* in the menu *Project* is selected, the *HelloWorld* application will be compiled and created with the standard GNU GCC C/C++ Compiler. You can find the executable in *workspace/ myHelloWorld/Debug*.

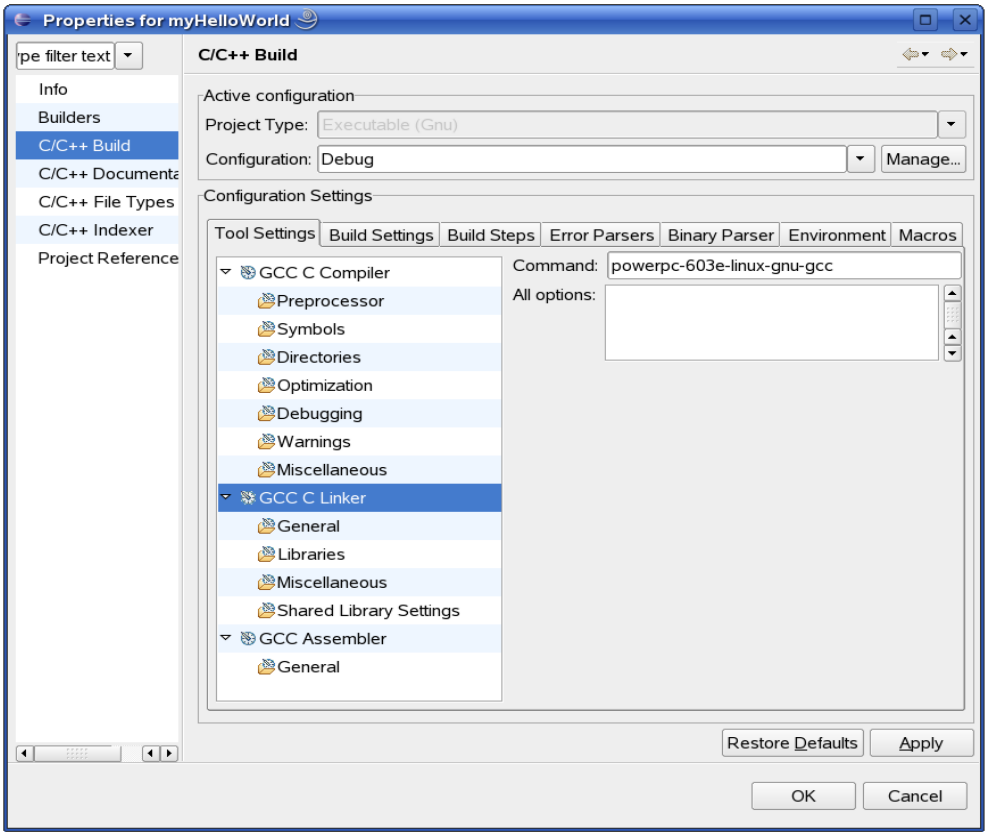
To compile your project for the target, you will have to use the GNU C/C++ Cross-Compiler.

- Right-Click the *myHelloWorld* project and choose Properties.

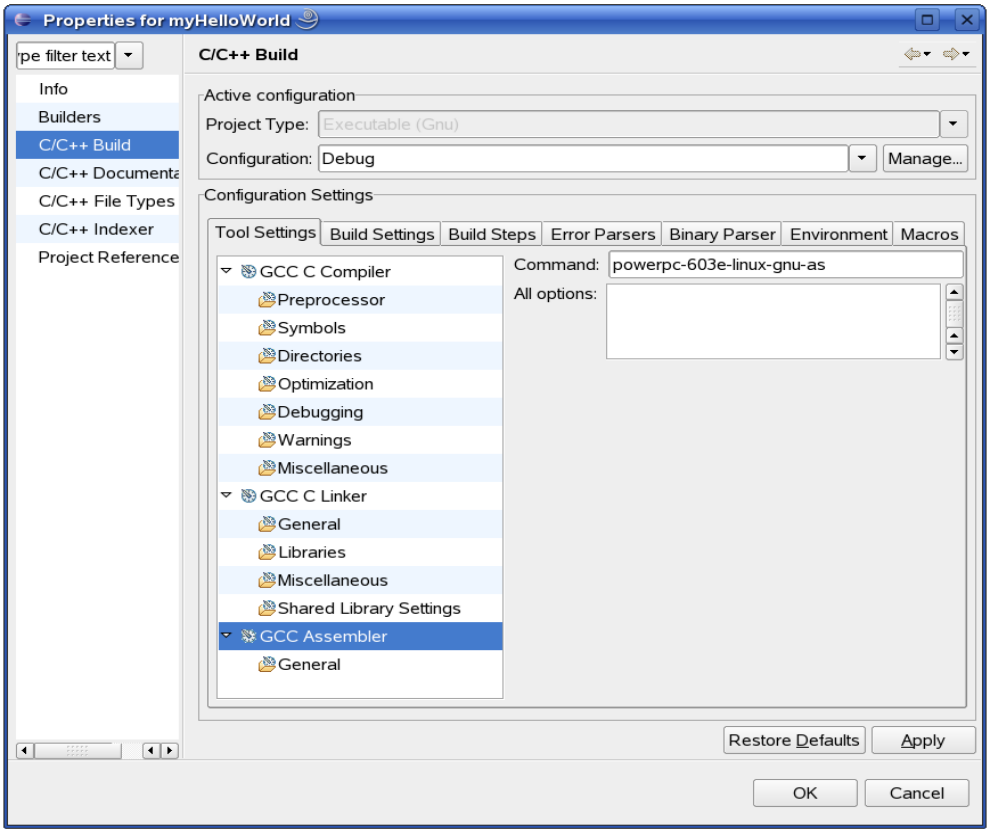
The properties dialog opens.



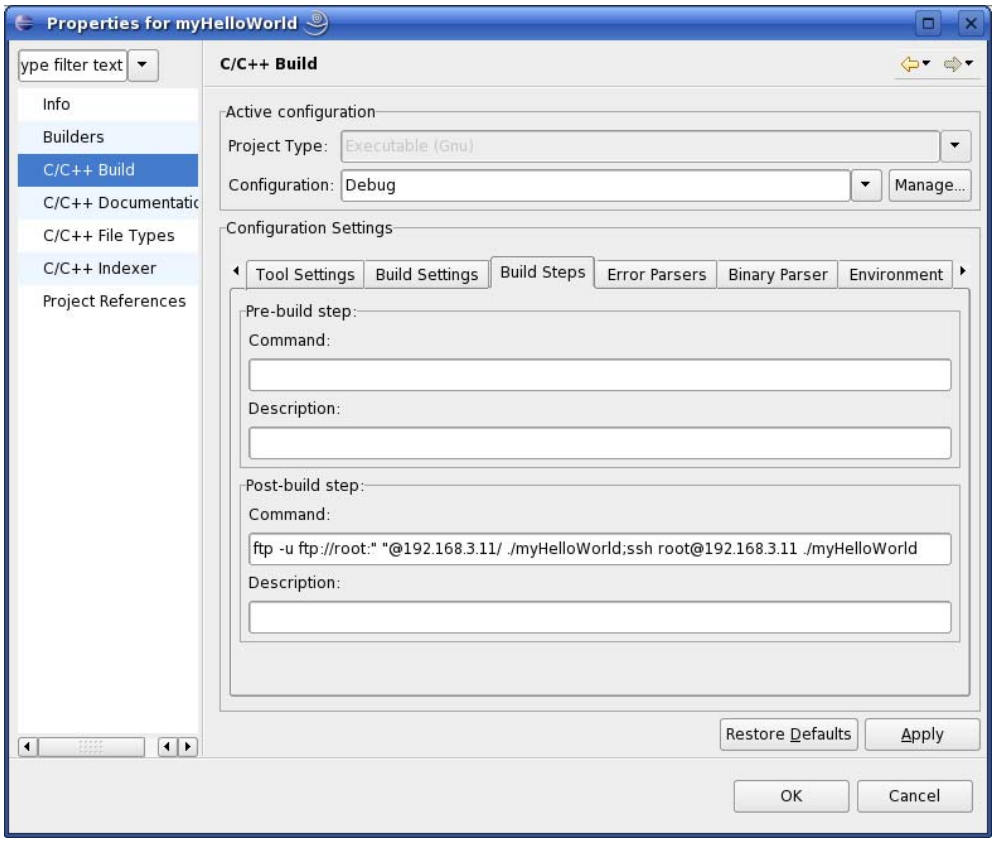
- Select *C/C++ Build*.
- Enter at Command: **powerpc-603e-linux-gnu-gcc.**



- Select *GCC C Linker***.**
- Enter at Command: **powerpc-603e-linux-gnu-gcc.**



- Select *GCC Assembler*.
- Change Command **as** to **powerpc-603e-linux-gnu-as**.
- Click on *Apply*.

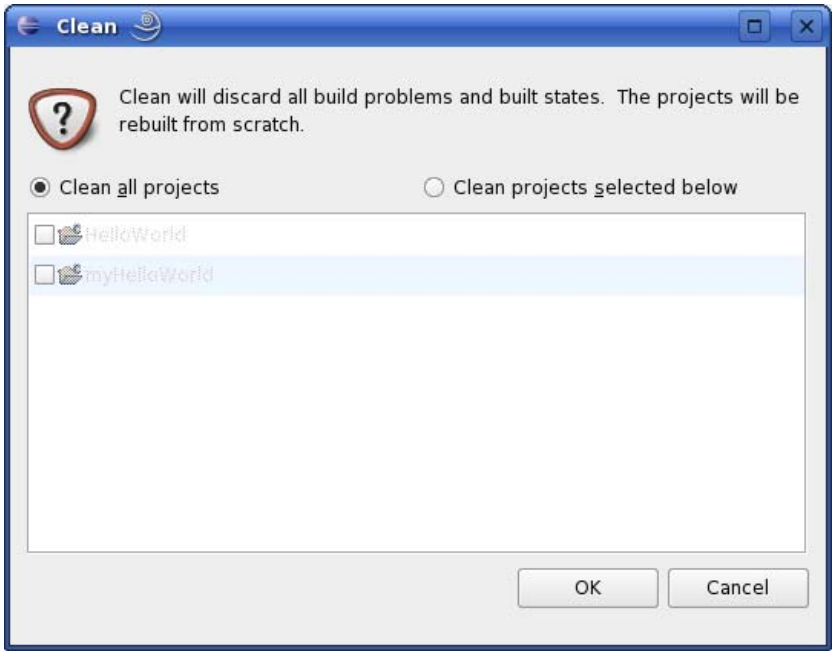


- Select the Build Steps tab:
- Enter following commands in the Command text field:

  ftp -u ftp://root:root@192.168.3.11/ ./myHelloWorld;
  ssh root@192.168.3.11 ./myHelloWorld

Be sure to enter the semicolon between ./myHelloWorld and ssh.

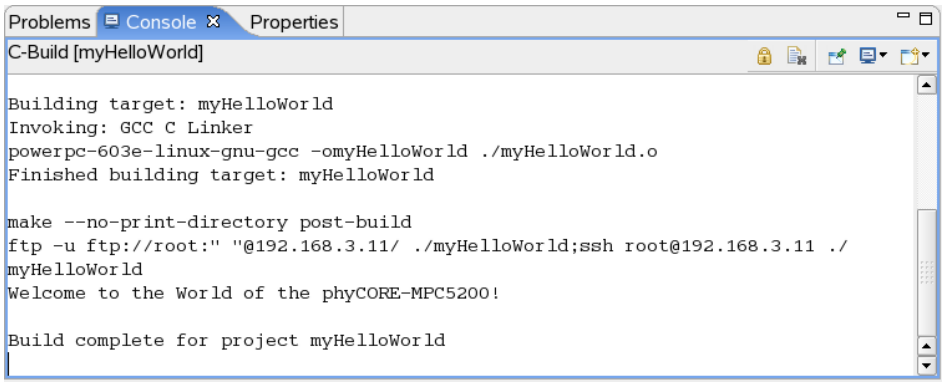
- Click on *Apply***.**
- Click on *OK***.**



- Select in the menu bar *Project->Clean*.
- Confirm with *OK*.

  The project will be rebuilt.
- Select the *Console* tab.

If no errors occur while building, you will see the following output:



You have successfully created your first own project with the Eclipse IDE. You have configured the project to create an executable program for your target platform.

3.5 Changing the Demo

Now we will extend the *myHelloWorld* application. The extended *myHelloWorld* application will write an output on the serial interface ttyPSC0 and the standard output.

- Open Eclipse if is not opened yet.
- Double-Click *HelloWorld.c* in the *myHelloWorld* project.

- First include two additional header files:

```
#include <unistd.h>
#include <fcntl.h>
```

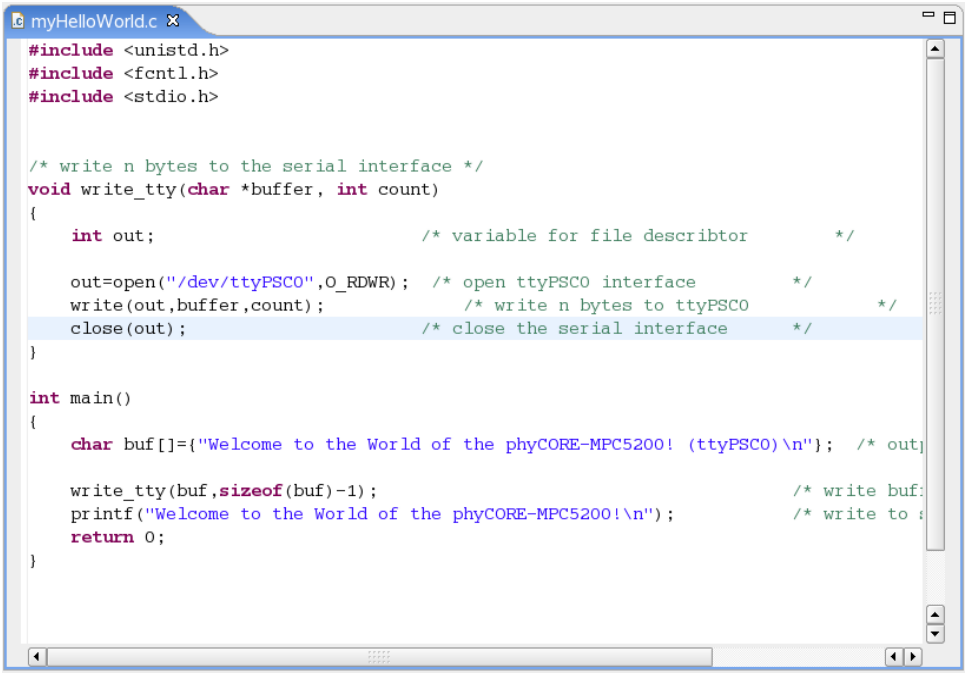
- Then add the function write_tty to write n bytes to the serial interface /dev/ttyPSC0.

```
void write_tty(char *buffer, int count)
{
int out;

out=open("/dev/ttyPSC0",O_RDWR);
write(out,buffer,count);
close(out);
}
```

- Enter the following two lines in the main function to declare the buffer und call the write_ttyS0() function.

```
char buf[]={"Welcome to the World of the phyCORE-MPC5200
        (ttyPSC0)\n"};

write_tty(buf,sizeof(buf)-1);
```

On the next screenshot you can see the complete program.



- Save your program after changing the code.

The Application will be compiled, built, copied to the target and executed.

Executing the program on the target with Komport:

- Click on the *Komport* icon on the desktop
- If your are not logged in, enter **root** and press *enter*.
- Type *./myHelloWorld* to start the application.

- You will see the following output:

  Welcome to the world of the phyCORE-MPC5200! (ttyPSC0)
  Welcome to the world of the phyCORE-MPC5200!

- Close Komport.

If you start the application over a ssh session you only see one output line. If you execute the program with Komport, you see two output lines.

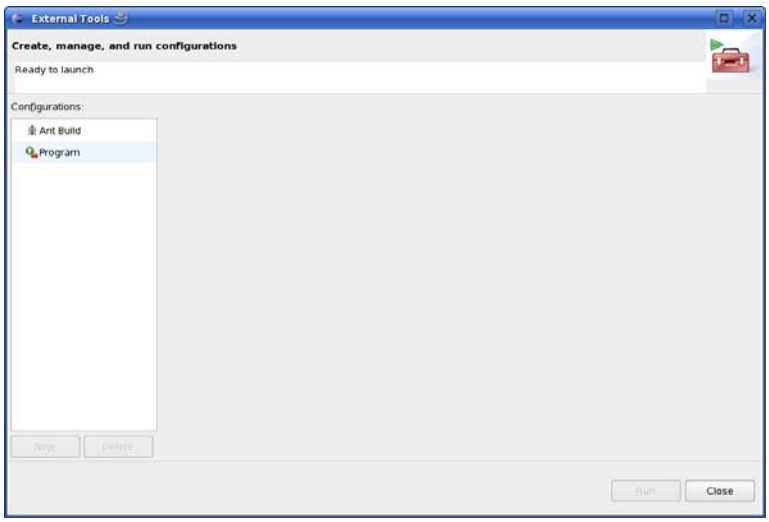
The first line is a direct output on the serial interface. You can't see this line in a telnet session, because you are connected over a TCP/IP connection to the target. With Komport you have direct access to serial interface, so you can see two output lines.

In this passage you have changed an existing application. In this part you also learned how to to access the serial interface. First you called the function *open()* on the device */dev/ttyS0*. The return value of this function was a file descriptor. With the file descriptor you called the function *write()* to send n bytes to the device */dev/ttyS0*. After that the file descriptor was closed with the function *close()*.

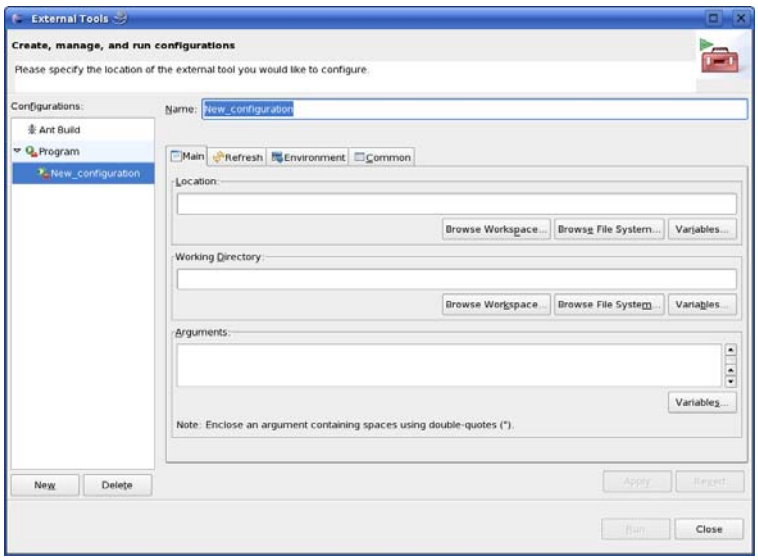
This principle procedure is quite typical for Linux, because Linux treats everything like a file.

3.6 Starting a program out of Eclipse on the target

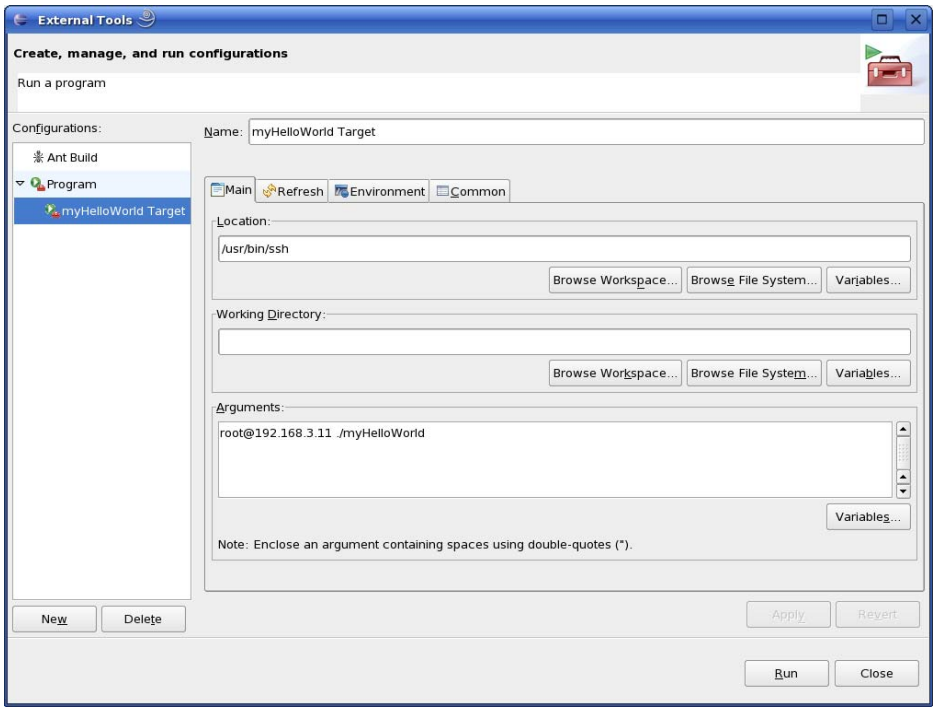
After compiling a project in Eclipse, the program is copied to target and directly executed. A program can also be executed on the target without compiling a project. In the following steps you will learn how to start a program on the target as external tool.



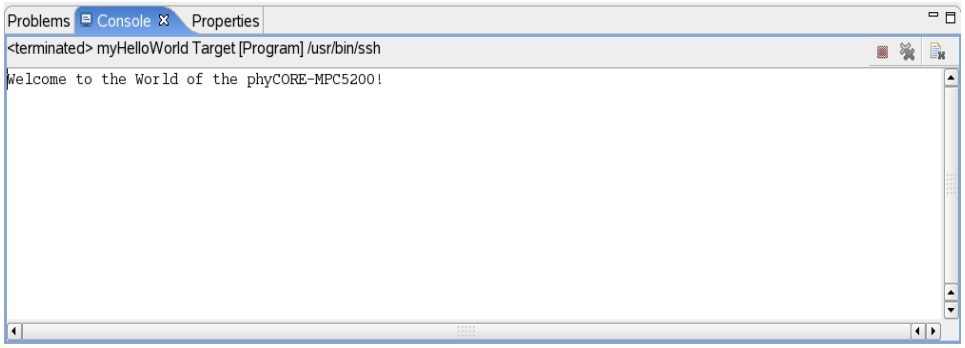
- Select in the menu bar *Run->External Tools->External Tools*



- Select *Program*.
- Select *New*.



- Enter *Name:* **myHelloWorld Target**
- Enter *Location:* **/usr/bin/ssh**
- Enter *Arguments:* **root@192.168.3.11 ./myHelloWorld**.
- Select *Apply*.



- Select *Run*.

If you want to execute the application the next time you can use the Run External Programs button from the menu bar.

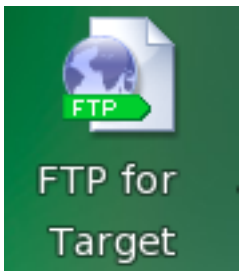
3.7 Starting the program when booting the target

In this passage you will integrate the created *myHelloWord* program into the start process of the target. When you have finished this part, the *myHelloWorld* will be started automatically each time you are starting the target.

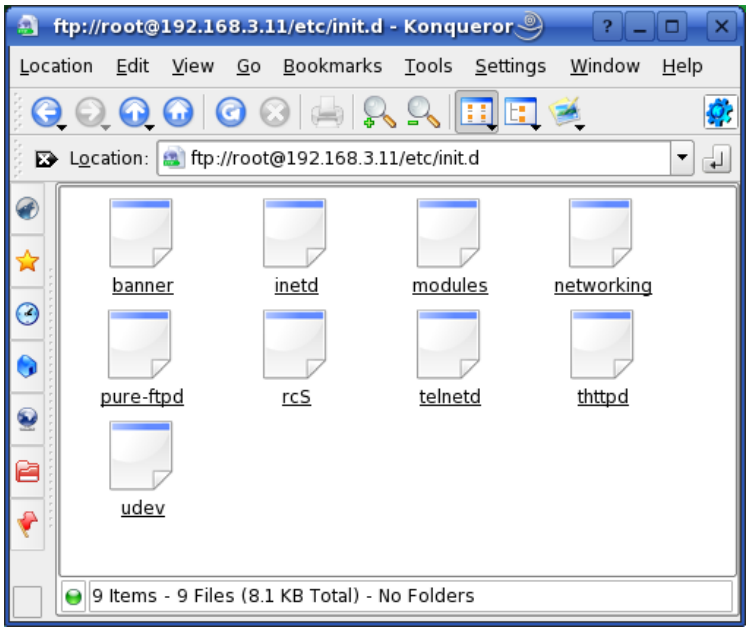
The scripts for controlling the system are placed in */etc/init.d*. These are executed directly or indirectly by */sbin/init*, the father of all processes. The configuration of the */sbin/init* is given by */etc/inittab*.

After system startup, */sbin/init* will switch on the default run level given in */etc/inittab*. It calls the run level master script /etc/init.d/rcS to start or stop services provided by the other scripts in /etc/init.d/. This is done by the help symbolic links in the directory /etc/rc.d/ to the directory /etc/init.d, which include the actual scripts.

First you will have to create a start script in */etc/init.d.*



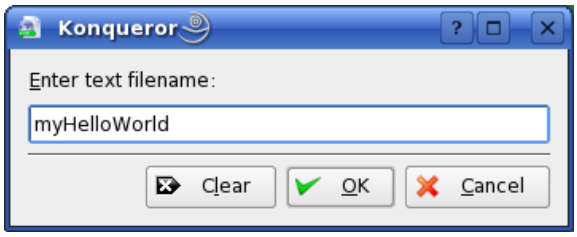
- Click on the Icon *FTP for target* on your desktop.



- Browse to */etc/init.d*. If an authorization dialog opens, click on *OK*.

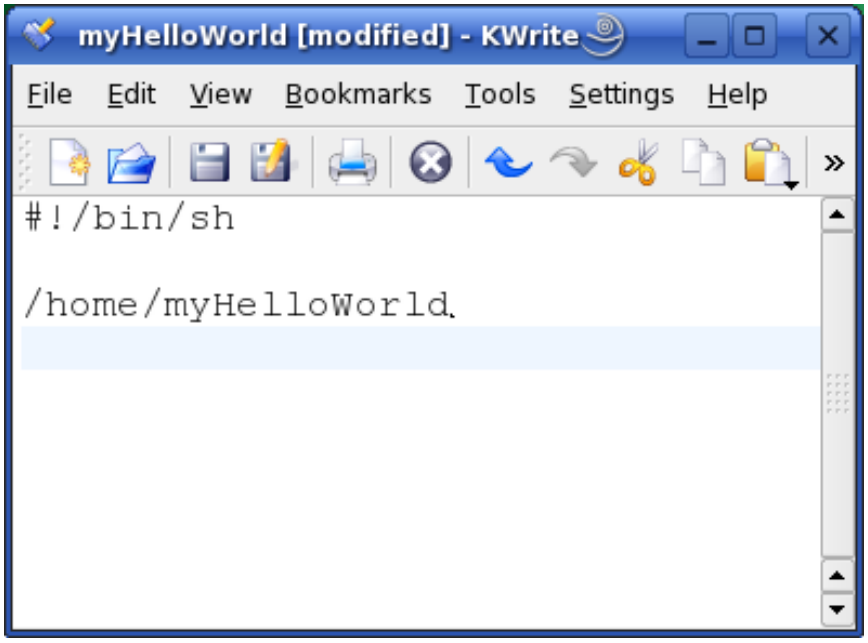
In the directory /etc/init.d you can find existing scripts.

- Right-Click in the opened window and select *Create New–>Text File.*



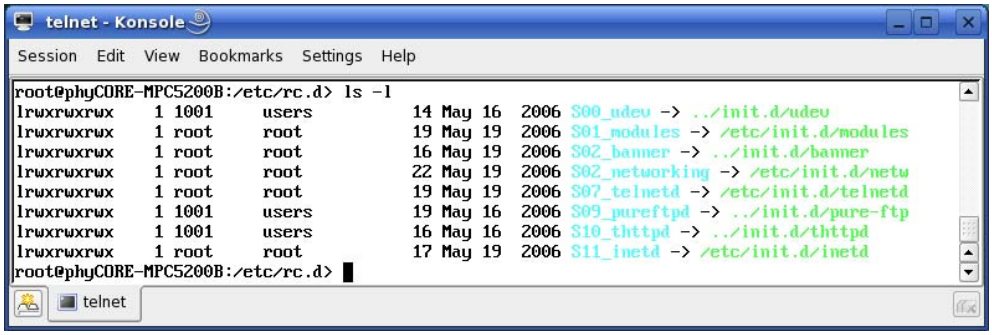
- Enter *myHelloWorld*.
- Click on *OK*.
- *Right-Click* on *myHelloWorld* and select *Open With*.
- Enter **kwrite** and click on *OK*.

The text editor kwrite starts with an empty document.



- Enter the following two lines:
  #!/bin/sh
  /home/myHelloWorld
- Select File –> Save
- Close the kwrite window.
- Close the FTP window.

- Click on the icon *Telnet for Target*



- Enter **root** and press **Enter** to login.
- Change to the directory */etc/rc.d*. Type the following command:

  cd /etc/rc.d
- Enter **ls -l** to show the directory content.

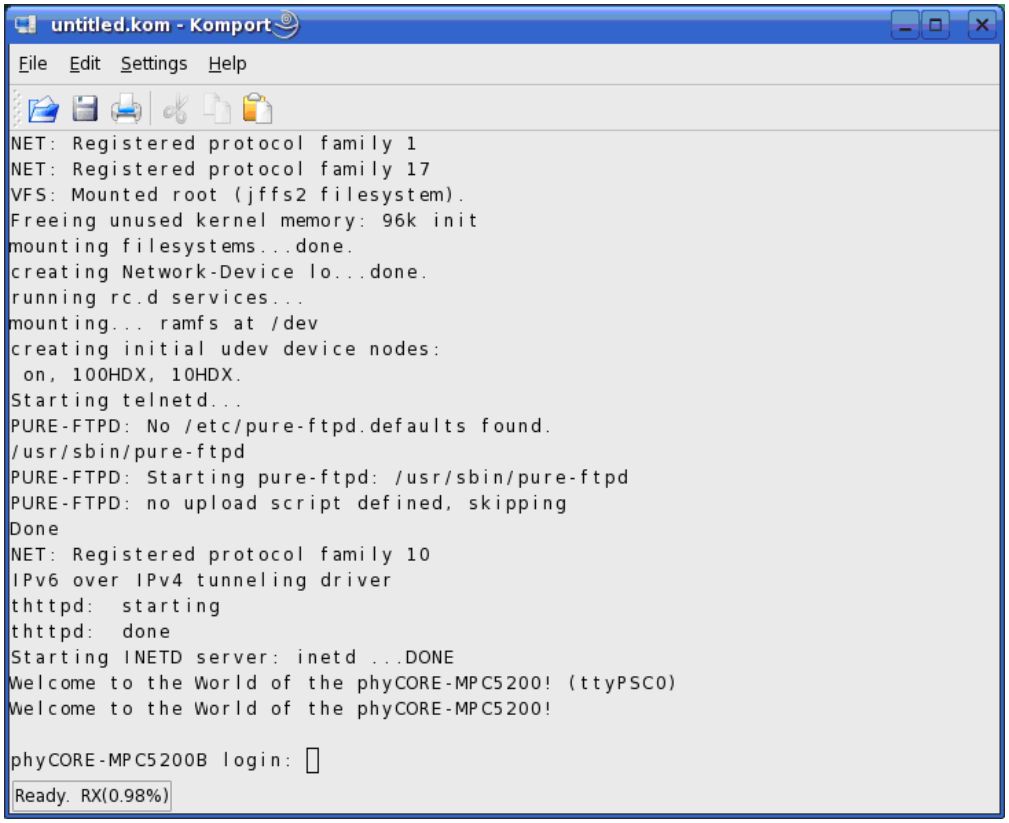
You can see the different links to the scripts in the directory */etc/init.d*. The script *udev* is the script which is started first, because the link starts with S00_ . The last started script is pwm. To start your program myHelloWorld automatically you have to create a new link to the start script.

On your target system could exist more than 12 scripts. If you want to create a new link, choose the next free number. In this example you can use S12_ …

- Create a symbolic link in /etc/rc.d to etc/init.d/myHelloWorld. Enter the following commands:

ln -s /etc/init.d/myHelloWorld /etc/rc.d/S12_myHelloWorld

- Type ls –l again to see the newly created link.
- Close the window.
- Open Komport.
- Push the RESET button on the target to restart your system.



The program myHelloWorld starts automatically on startup.

- Close Komport.

Now you can add own programs in the root file system and start these programs automatically.

You have successfully passed the Getting Involved part of the QuickStart.

4 Debugging an Example Project

20 min

In this chapter you will learn using the GNU GDB-Debugger on the Host for Remote Debugging in conjunction with the GDB-Server on the target. The GNU GDB debugger is the symbolic debugger of the GNU project and is arguably the most important debugging tool for any Linux system.

First you will start the GDB-Server on the target. Then you will configure the Eclipse Platform and start the GDB-Debugger out of Eclipse using the Debug View.

The CDT extends the standard Eclipse Debug View with functions for debugging C/C++ code. The Debug View allows you to manage the debugging or running of a program in the Workbench. Using the Debug View you will be able to set breakpoints/watchpoints in the code and trace variables and registers. The Debug View displays the stack frame for the suspended threads for each target you are debugging. Each thread in your program appears as a node in the tree, and the Debug View displays the process for each target you are running.

The GDB is running on the host and used to debug. The GDB-Server is running on the target and it is used to start and control the program to debug. The GDB and GDB-Server can communicate over TCP/IP and the serial interface. In this QuickStart we will only describe debugging via TCP.

4.1 Starting the GDB-Server on the target

In this passage you will learn how to start the GDB-Server on the target. The GDB-Server will be used to start the myHelloWorld program. To debug a program with the GDB, the program needs extending debugging symbols. This has already been added while building the program.

- Open Komport

```
phyCORE-MPC5200B login: root


root@phyCORE-MPC5200B:~> gdbserver 192.168.3.10:10000 myHelloWorld
Process myHelloWorld created; pid = 703
Listening on port 10000
```

- Type **root** and press *Enter*.
- Start the gdbserver:

  gdbserver 192.168.3.10:10000 myHelloWorld

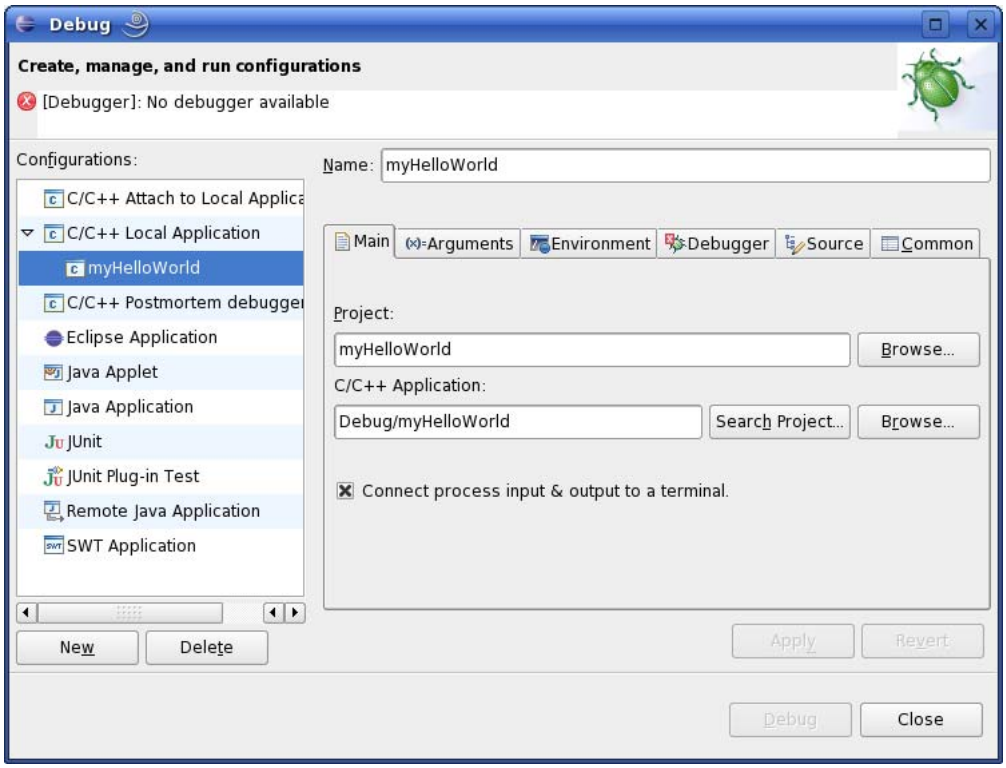
You have started the GDB-server on the target. The GDB-Server is now waiting for connections on port 10000.

4.2 Configuring and starting the debugger in Eclipse

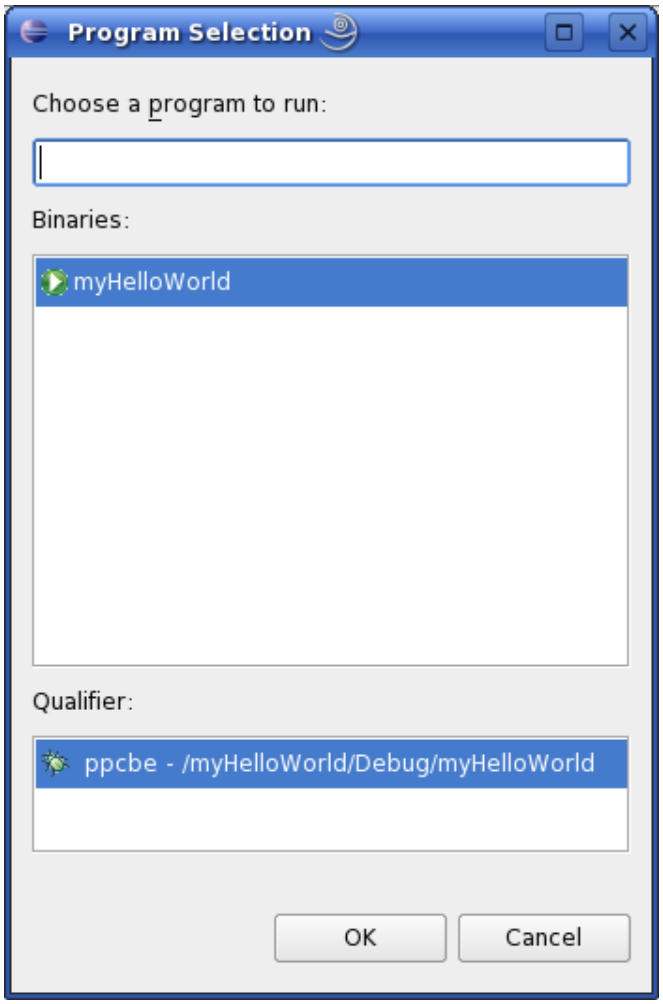
In this passage you will learn how to configure your project settings to use Eclipse with the GNU GDB debugger. After the configuration of your project settings, the GDB debugger will start and connect to the GDB-server on the target.

- Start Eclipse if the application is not started.
- Select *myHelloWorld* in the Navigator window.
- Select in the menu bar *Run->Debug*.

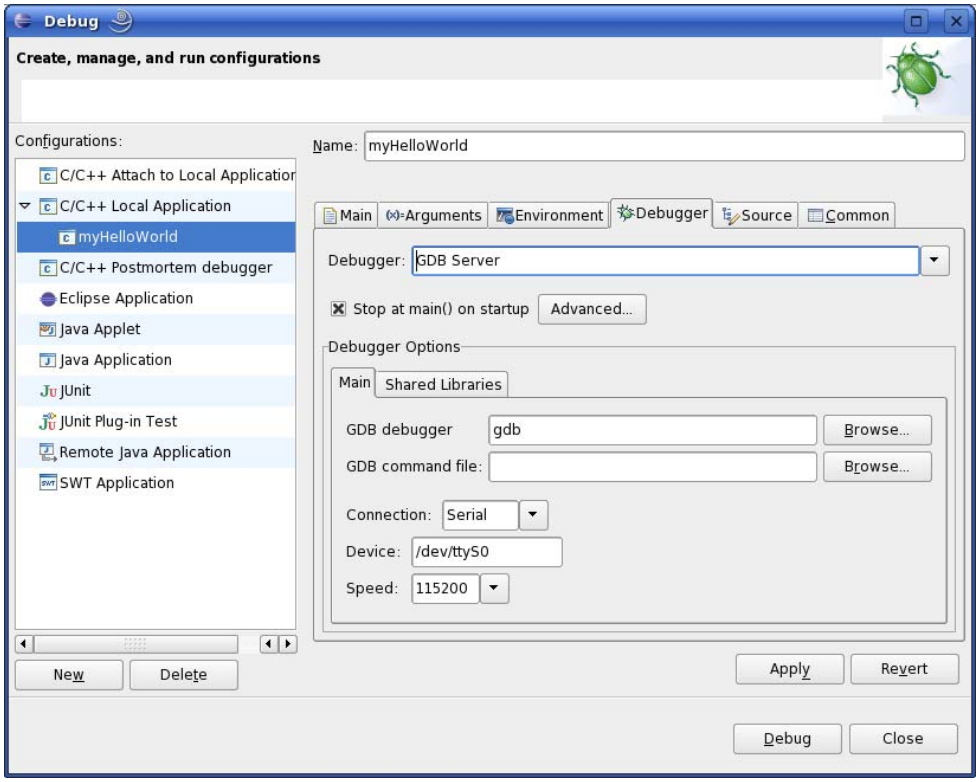
A dialog to create, manage and run applications will appear.



- Select *C/C++ Local Application*.
- Click on *New.*



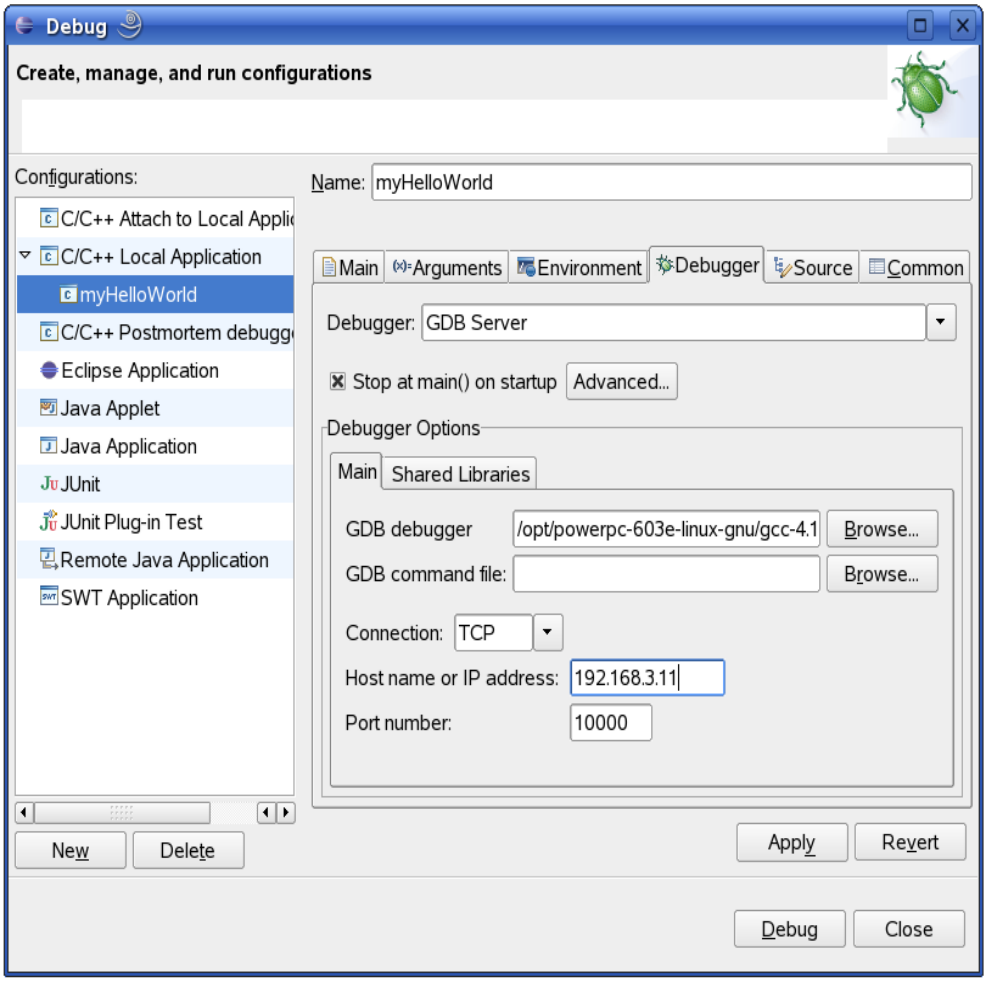
- Select the *Search Project* button.
- Click on *OK* .



- Select the *Debugger* tab.
- Select Debugger: *GDB Server*.
- Click on the *Browse* button in the line of the GDB debugger.

A new dialog to choose the directory of the GDB opens.

- Double-Click on *File System.*
- Navigate to */opt/powerpc-603e-linux-gnu/gcc-4.1.2-glibc-2.5-kernel-2.6.18/bin.*
- Select *powerpc-603e-linux-gnu-gdb.*
- Select *OK.*



- Select Connection: *TCP*
- Enter the Host name IP address: **192.168.3.11**
- Click the *Apply* button.

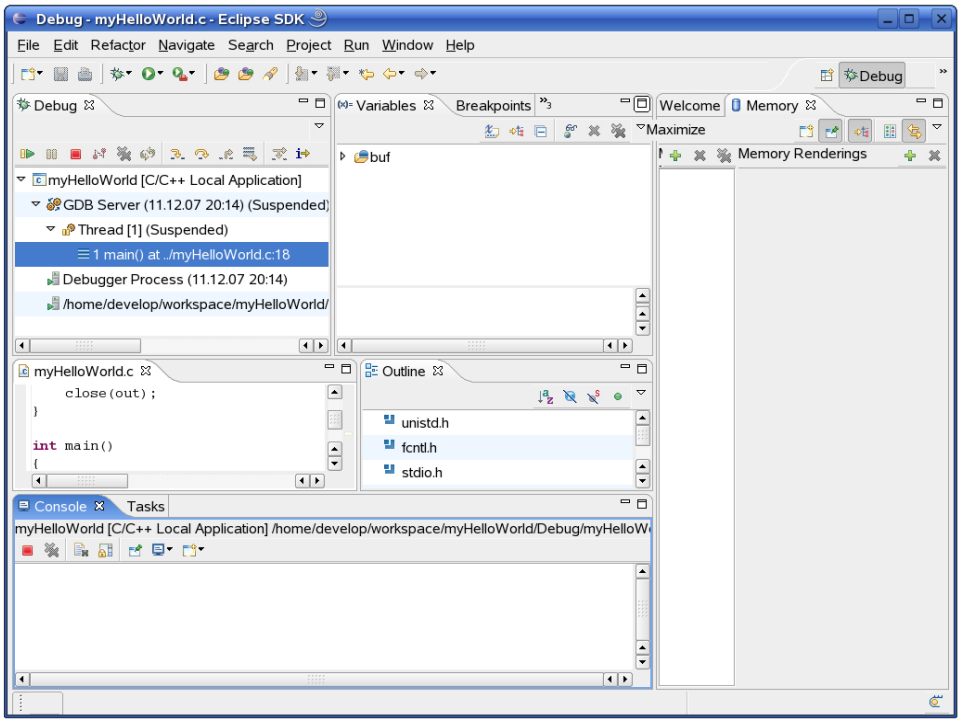
The host name IP address, is the IP address of the target.

- Click *Debug* button.

A new dialog opens.

- Select *yes* to switch to the debug perspective.

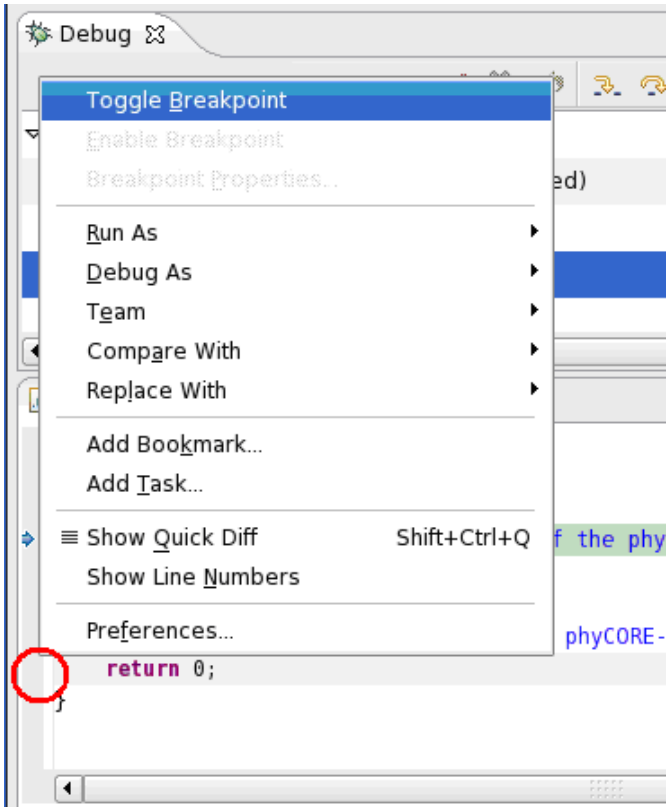
The debug perspective opens and the debugger stops at the first line automatically. The GDB is connected to the GDB Server on the target.



You have configured your project for remote debugging. You have started the GDB debugger in Eclipse and connected the GDB with the GDB Server. You can start to debug the project.

4.3 Setting a breakpoint

Now you will set a breakpoint in your program. The breakpoint will be set in the last line of the main function. If you resume the application, the debugger will stop in this line.

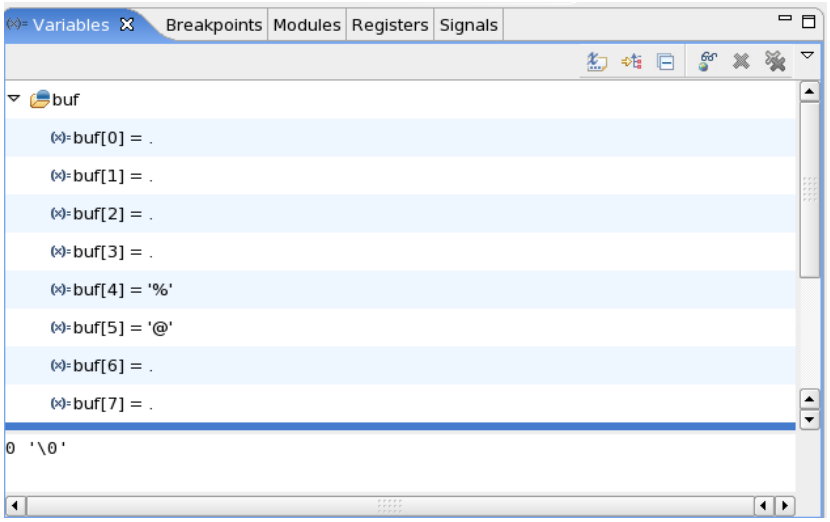


- Select the last line in main.
- *Right-Click* left in the small grey border and select *Toggle Breakpoint* to set a new breakpoint.

4.4 Stepping and Watching Variables Content

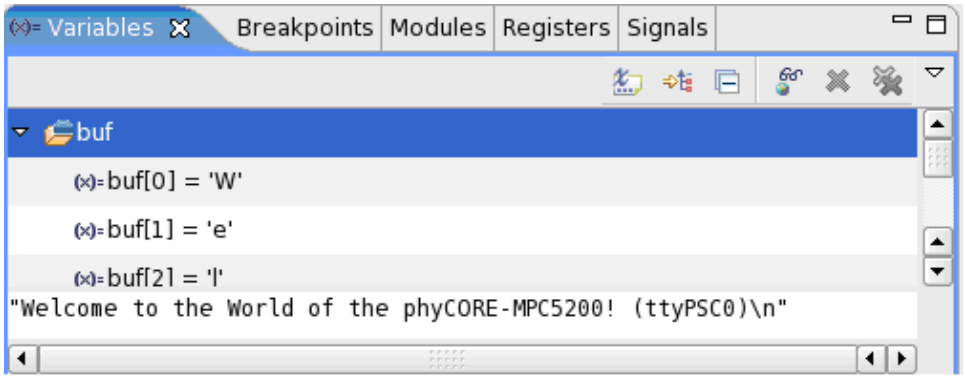
In this part you will step through the example project with the debugger. You will also learn how to watch the content of a variable.

- Expand *buf* in the variables window

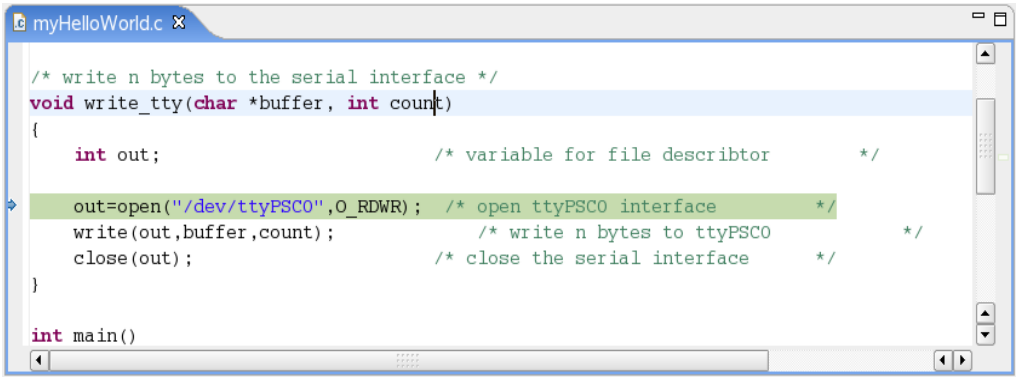


- Click on the *Step Over*  button in the Debug Window to step to the next line.

You will see the content of *buf* in the Variable Window.



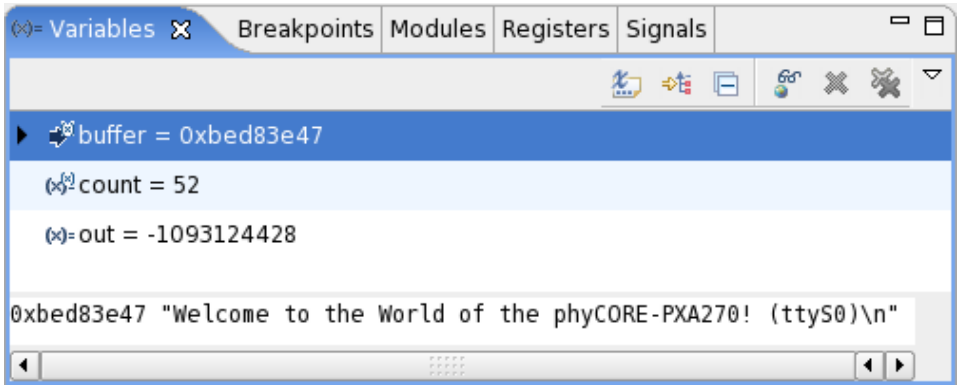
- Click on the variable buf.



- Then click on the button *Step into*  to enter the function *write_tty*.

The debugger stops in *write_tty*.

You will see the following variable window:

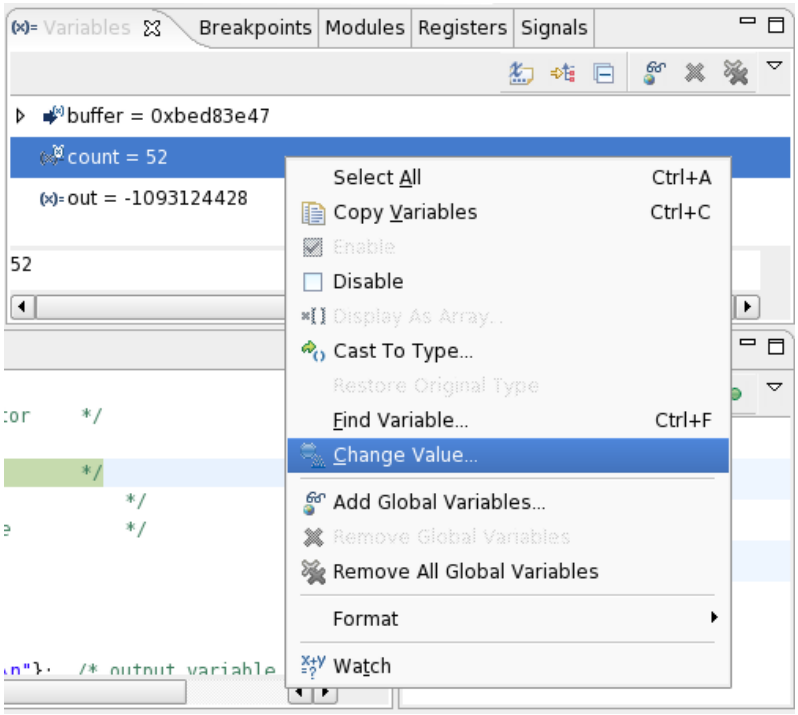


- Click on the variable buffer.

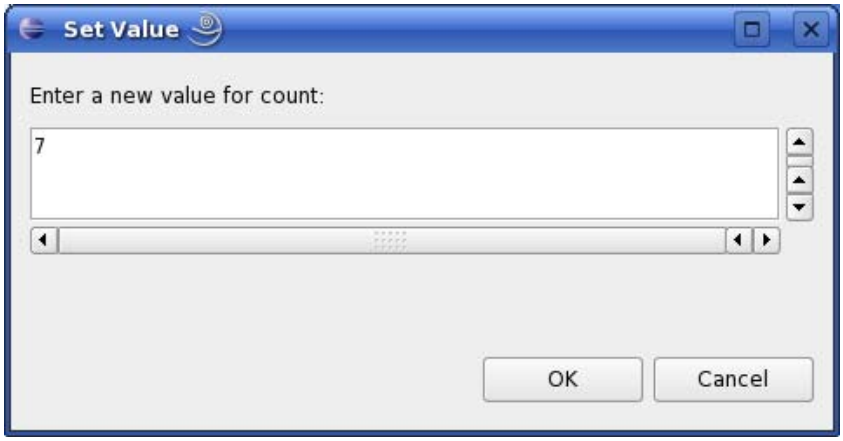
There will probably be another address at the pointer *buffer*.

4.5 Changing Variables Values

In this section you will change the value of a variable. At the end of this part you will see the effect of the changes.

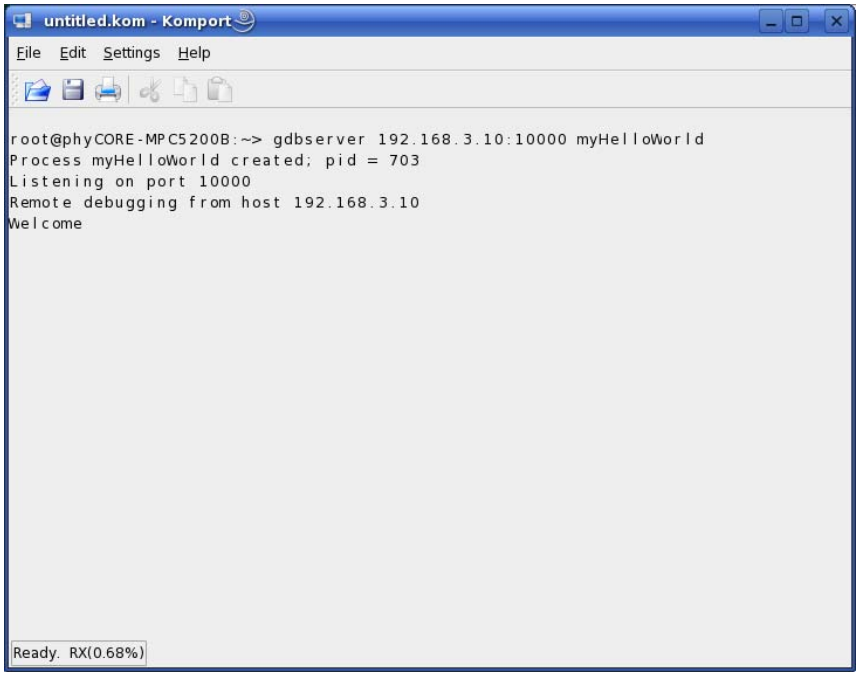


- Select *count* in the variable window.
- Right-click on *Count* and select *Change Value*.



- Change the value of count to **7** and select *OK***.**

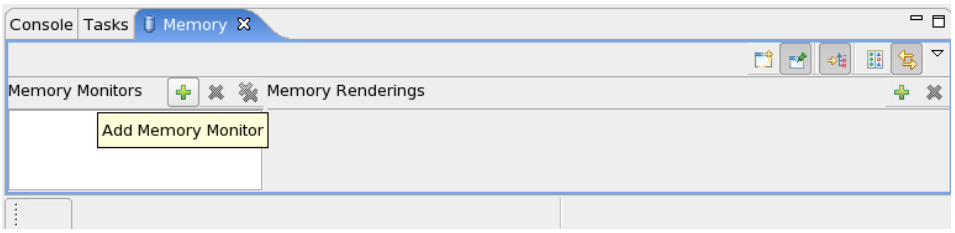
- Open Komport if the application is not open.
- Chang to Eclipse.
- Click on the *Step Over*  button for two times.
- Change to Komport.



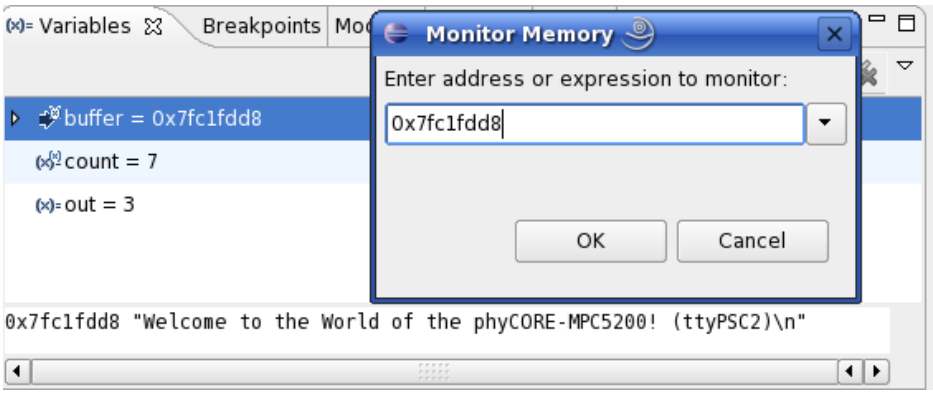
You will see the output *Welcome* in the Komport window.

4.6 Using the Memory Monitor

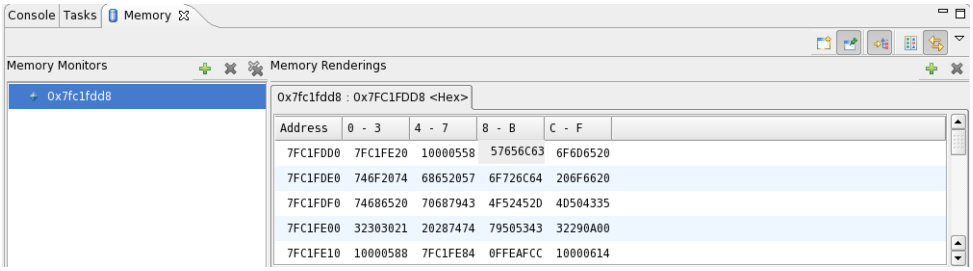
At the last part in this chapter you will use the memory monitor to watch the content at a memory address.



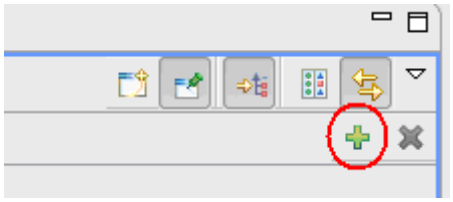
- Select the *Memory* tab in the window below.
- Click on the button *Add Memory Monitor*.



- Enter the address of *buffer* and click on *OK*.

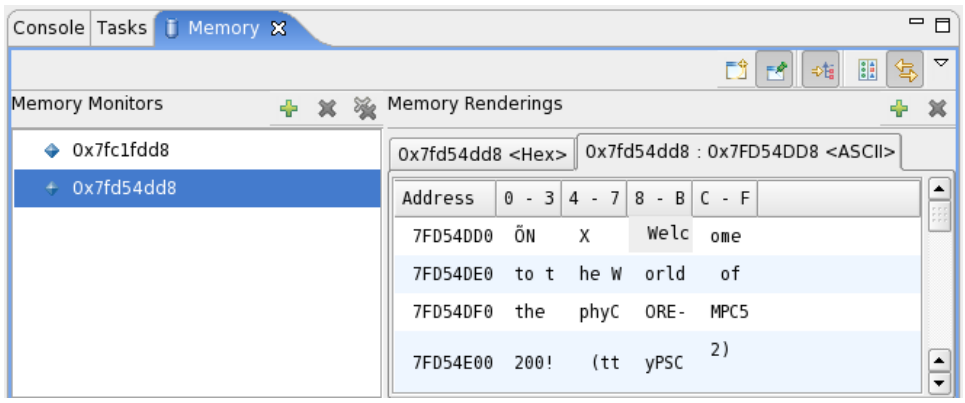


- Change the window size.



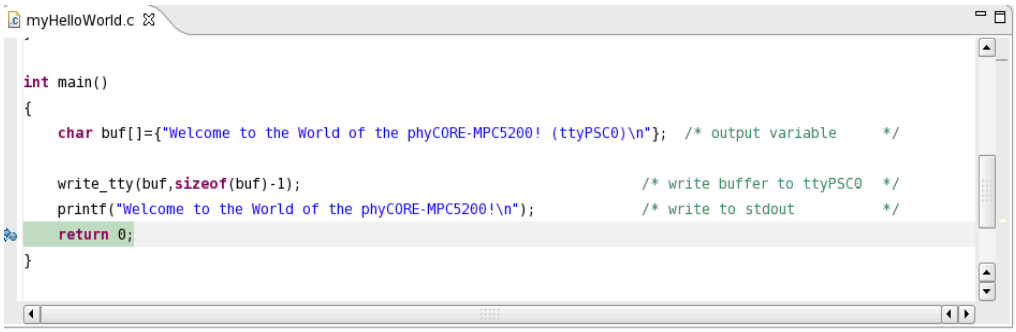
- Click on the *Add Rendering Button*.

- Select ASCII and click on the *OK* button.



You can see the content of the variable buf at the address 0x7fc1fdd8.

- Next click on the *Resume* button in the menubar.



The debugger stops at the breakpoint in the line of return.

- Click on the *Resume* button to end the application.

You have successfully passed the debugging chapter. You are now able to configure and use Eclipse for remote debugging. You can step through a project, watch and change the content of variables and you can use the memory monitor to view the content at a memory address.

5 Further Information

In the PTXdist User Manual you can find further information. You can find the manual in the directory */doc/OSELAS* on your setup cdrom.

The PTXdist User Manual includes information about the following topics:

- Installation and Configuration of PTXdist
- Using and Building a Toolchain
- Create and activate a project
- Running phyCORE-MPC5200 from network only
- Running phyCORE-MPC5200 stand alone
- U-Boot and phyCORE-MPC5200
- phyCORE-MPC5200's BSP
- Using CAN on phyCORE-MPC5200

6 Summary

This QuickStart Instruction gave a general "Rapid Development Kit" description, as well as software installation advice and an example program enabling quick out-of-the box start-up of the phyCORE®-MPC5200B in conjunction with the Eclipse IDE and GNU GCC/C++ software tools.

In the Getting started section you learned to configure your host to provide a basis for working with your target platform. You installed the Rapid Development Kit software and you learned to copy and run a program on the target.

In the Getting More Involved section you got a step-by-step instruction on how to configure and build a new kernel, modify the example, create and build new projects and copy output files to the phyCORE –MPC5200B using Eclipse.

The Debugging part of this QuickStart gave you information on setting up and using the GDB debugger with the Eclipse IDE. You learned how to set breakpoints, watching and changing variables content and using the memory monitor.

Document: **phyCORE®-MPC5200B IO with Linux QuickStart Instructions**
Document number: **L-679e_6, August 2009**

How would you improve this manual?

Did you find any mistakes in this manual? page

Submitted by:
Customer number: _____

Name: _____

Company: _____

Address: _____

_____

Return to:

PHYTEC Messtechnik GmbH
Robert-Koch-Str. 39
55129 Mainz
Fax: +49 (6131) 9221-26

Published by

PHYTEC

Ordering No. L-679e_6
Printed in Germany